AD-A038 140 ROYAL AIRCRAFT ESTABLISHMENT FARNBOROUGH (ENGLAND) F/G 22/3
THE ELLIPTIC ORBIT INTEGRATION PROGRAM POINT.(U)
SEP 76 G E COOK, M D PALMER

UNCLASSIFIED RAE-TR-76129 DRIC-BR-55559 NL

1 OF 1
AD
A038140

END
DATE
FILMED
4-77

AD A038140

BR55559

UNLIMITED

TR 76129

SEPTEMBER 1976

ROYAL AIRCRAFT ESTABLISHMENT

TECHNICAL REPORT 76129

1

NW

COPYRIGHT ©

CONTROLLER HMSO LONDON

1976

THE ELLIPTIC ORBIT INTEGRATION PROGRAM POINT

by

G. E. Cook

M. D. Palmer

DDC

RECEIVED

APR 7 1977

D

[redacted]

DDC FILE COPY

UNLIMITED

UDC 521.6 : 629.19.085 : 629.195 : 621.396.946 :
629.195 : 521.26 : 517.391 : 518.5

ROYAL AIRCRAFT ESTABLISHMENT

9 Technical Rep[redacted]t, 76129

Received for printing 17 September 1976

6 THE ELLIPTIC ORBIT INTEGRATION PROGRAM POINT.

by

10 G. E. Cook
M. D. Palmer

11 Sep 76

14 RAE-TR-76129

12 95p

SUMMARY

POINT is a computer program which evaluates the development of an earth satellite orbit by numerical integration. Provision is made for the inclusion of the following perturbations:

(i) earth's gravitational potential,

(ii) atmospheric drag, and

(iii) the gravitational attractions of the sun and moon.

A detailed description of the program is given, with full instructions for its use.

18 DRIC

19 BR-55559

DDC
RECEIVED
APR 7 1977
D

White Section
Buff Section
AVAIL. and/or SPECIAL
A

Departmental Reference: Space 512

310450

CONTENTS

1 INTRODUCTION

POINT is the acronym for a computer program for orbit integration. The program, which numerically integrates the equations of motion of an earth satellite in a highly eccentric orbit, was originally conceived as a component of the computer software required for the analysis of the launch phase of a synchronous satellite mission. The program was written with a view to more general application, however, and will be useful in pre-launch analyses of other satellite projects. A shortened version has already been used in this role to assess the orbital stability of the proposed Astronomical Roentgen Observatory (ASRO)[1].

Given a set of initial conditions, the program generates ephemerides of an earth-orbiting satellite. At the epoch, the orbit can be specified either by the cartesian components of the geocentric position and velocity vectors, by a standard set of launch vehicle injection conditions or by one of three types of orbital elements. The ephemerides, of which any combination may be obtained, consist of the cartesian components of geocentric position and velocity, osculating orbital elements, look angles for a maximum of four ground stations and a table of apses.

Orbit development is obtained by the numerical integration of the equations of motion as formulated in the Cowell method. Large variations occur in the satellite's velocity and in the perturbing forces around a highly eccentric orbit. At perigee where the velocity is high, and the forces are changing rapidly, a small integration step length is required. Use of this step length around the entire orbit is unnecessary and inefficient, however. Analytical step regulation[2] is therefore used to obtain the required accuracy without loss of efficiency.

Luni-solar perturbations caused by the differential attractions of the sun and moon are computed as required during integration. The sun-moon coordinates are stored at daily intervals in a disc file. The coordinates originate from the JPL ephemeris tapes[3,4] and have been transformed into PROP axes (see section 4). Although solar radiation pressure is not incorporated at present, the program is written in a manner which will enable it to be inserted at a future date. The asphericity of the earth is taken into account approximately by including the effects of zonal harmonics up to J_9 and tesseral harmonics up to $J_{4,4}$. Atmospheric drag is included with the density evaluated from a modified form[5] of the Jacchia 1965 model atmosphere.

POINT is written in 1900 FORTRAN for use on ICL 1900 series computers. The program requires approximately 20K words of core store, and consists of a main program and 33 subprograms. The program units are listed in Appendix A, while the calling structure is illustrated in Fig.1. A flow chart for the main program is given in Fig.2. The subprogram specifications are given in Appendix B.

2 PROGRAM FUNCTION

The program may take as input, at a given epoch, either the geocentric cartesian components of the position and velocity vectors $(x,y,z,\dot{x},\dot{y},\dot{z})$, or a standard set of launch vehicle injection conditions or one of three types of orbital elements. The launch vehicle injection conditions consist of speed, climb angle, azimuth of the velocity vector, radial distance, geocentric latitude and longitude. The three types of orbital elements are: firstly, a set of osculating elements defined by the instantaneous position and velocity vectors, and consisting of semi major axis a, eccentricity e, inclination i, right ascension of the ascending node Ω, argument of perigee ω and mean anomaly M; secondly, a set of two-line elements in the form issued by the USAF Space Defense Center (SDC); and thirdly, a set of RAE five-card elements produced as output by the computer program PROP[6]. The latter are mean elements in the sense defined by Kozai[7], i.e. time averages per revolution of the osculating elements.

The program generates up to three types of ephemerides each of which is a table of 'satellite positions' at uniform intervals of time. For the purpose of this Report 'satellite position' means either the cartesian components of the geocentric position and velocity vectors, a set of osculating orbital elements, or a set of look angles for a given ground station. Look angles consist of azimuth A, measured clockwise from true north, elevation E, range and range rate. The program can also determine the satellite's position and time at an apse; this information is then output in tabular form. Although only three types of ephemerides can be printed at present, it would be straightforward to increase the program's flexibility by including other types.

For cartesian components it is possible to obtain the covariance matrix as an additional output provided that the covariance matrix of the initial conditions is available as part of the input. A word of caution is needed here, however. Since the propagation of errors by means of a covariance matrix is basically a linear concept, difficulties can be encountered if the initial

errors are large, e.g. as may result from the errors associated with a set of launch vehicle injection conditions.

3 EQUATIONS OF MOTION AND INTEGRATION PROCEDURE

For an earth satellite the equations of motion may be expressed in vector form as

$$\ddot{\underline{r}} = -\mu r^{-3}\underline{r} + \underline{F} , \tag{1}$$

where $\underline{r}$ is the position vector relative to the earth's centre of mass, μ (= GM) is the gravitational constant of the earth, and $\underline{F}$ is the perturbing acceleration. Orbit development is obtained by the numerical integration of the equations of motion as formulated in the Cowell method. Because of the large variations in the velocity and perturbing accelerations around a highly eccentric orbit, a constant time interval cannot be used if the computation is to be efficient. To overcome this difficulty, the equations are transformed in such a way that a constant step-length can be used, i.e. analytical step regulation is introduced. Time t is replaced by the independent variable s, defined by

$$dt/ds = r^{k}/\kappa , \tag{2}$$

where κ is a constant. Merson has discussed[2] the selection of k and κ to give the best results and suggested the use of $k = 3/2$ and $\kappa = \mu^{1/2}$; these values are used in the program. On changing to the independent variable s, where

$$t' = dt/ds = \mu^{-1/2} r^{3/2} , \tag{3}$$

the equation (1) becomes

$$\underline{r}'' = -\underline{r} + \frac{3}{2}\left(\frac{\underline{r} \cdot \underline{r}'}{r^2}\right)\underline{r}' + \frac{r^3}{\mu}\underline{F} . \tag{4}$$

The equation for t' can be differentiated to give

$$t'' = \frac{3}{2}(\underline{r} \cdot \underline{r}')/(\mu r)^{\frac{1}{2}} , \tag{5}$$

so that we have four second-order differential equations (for x'',y'',z'',t''). The integration is based on an eighth-order Gauss-Jackson second-sum process.

A sixth-order Butcher process is used to set up the difference table required before the second-sum procedure can be started.

4 TIME AND COORDINATE SYSTEMS

Calendar dates are reckoned in Modified Julian Days (MJD), which are related to (ordinary) Julian Days by the formula

$$MJD = JD - 2400000.5 \quad .$$

The coordinate system used internally is the one suggested by Kozai[8]. The origin O is at the earth's centre of mass and the Oz axis points towards the north pole. Ox lies in the plane of the true equator of date, but instead of pointing towards the true equinox of date it is directed towards a projection of the mean equinox of the epoch 1950.0 (MJD 33281.9234); Oy completes the right-handed system Oxyz .

5 UNITS AND CONSTANTS

5.1 General

Distances are measured in kilometres (for input of a covariance matrix, however, distances may alternatively be given in feet). Angles are held as radians inside the computer, but are measured in degrees for both input and output. Time is measured in seconds, except when identifying specific instants or time intervals.

Constants used in the program are assigned only once, either in the main program or in the BLOCK DATA subprogram.

5.2 Geophysical constants

Values of the geophysical constants are assigned in the BLOCK DATA subprogram; current values are listed below, E followed by an integer designating an exponent. Values for the even zonal harmonics have been taken from Ref.9, while those for the odd zonal harmonics have been taken from Ref.10. Values for the tesseral harmonics have been taken from a set derived by Wagner[11].

Ccophysical constants

Constant		FORTRAN variable	Value
Earth's gravitational constant μ (km^3s^{-2})		EMU	398601.3
Mean equatorial radius of earth R (km)		ERAD	6378.163
Earth's zonal harmonics	J_2	EJ2	1082.637E-6
	J_3	EJ3	-2.531E-6
	J_4	EJ4	-1.619E-6
	J_5	EJ5	-0.246E-6
	J_6	EJ6	0.558E-6
	J_7	EJ7	-0.326E-6
	J_8	EJ8	-0.209E-6
	J_9	EJ9	-0.094E-6
Earth's tesseral harmonics	C_{22}	C22	2.4369E-6
	S_{22}	S22	-1.4005E-6
	C_{31}	C31	2.0192E-6
	S_{31}	S31	0.2278E-6
	C_{32}	C32	0.7783E-6
	S_{32}	S32	-0.7552E-6
	C_{33}	C33	0.7387E-6
	S_{33}	S33	1.4343E-6
	C_{41}	C41	-0.5175E-6
	S_{41}	S41	-0.4814E-6
	C_{42}	C42	0.3444E-6
	S_{42}	S42	0.7021E-6
	C_{43}	C43	1.0390E-6
	S_{43}	S43	-0.1192E-6
	C_{44}	C44	-0.1846E-6
	S_{44}	S44	0.2508E-6
Earth's angular velocity (rad/s)		EOMEGA	7.292115147E-5
Sun's gravitational constant (km^3s^{-2})		SUNMU	1327127E5
Moon's gravitational constant (km^3s^{-2})		SELMU	4902.756
Earth's precession rate (rad/s)		PREC	6.079E-12

6 PROGRAM DESCRIPTION

The program starts by reading all the input data and carrying out any necessary transformations. The latter define the orbit as the cartesian

components of the geocentric position and velocity vectors at epoch; if a covariance matrix is to be propagated, this may also have to be transformed so that its elements refer to position and velocity components. Certain flags are set and those constants which are a function of the input data are calculated.

A simple heading giving the perturbing forces included (i.e. air drag, luni-solar perturbations) is written to the line printer.

The integration period and interpolation interval are converted to days and the time of the first interpolation found relative to epoch. The integration, which may be forwards or backwards in time, is begun, and after each integration step the required interpolations are performed. The time of the first interpolation occurring after the latest step is then set. After each interpolation, according to the output options chosen, the interpolated position and velocity components may be written directly to the line printer (and if required to a permanent disc file) or transformed into osculating elements and/or range, range rate, azimuth and elevation for a given ground station. These latter forms are stored consecutively in an unformatted scratch disc file. The associated covariance matrix may also be written directly to the line printer.

At each integration step, the rate of change of the radius vector is calculated. This quantity changes sign at an apse enabling the integration steps on either side to be identified. Interpolations are performed to find the time of the apse and the corresponding osculating elements and radial distance. This information is then written to a second unformatted scratch disc file.

When the integration is complete, each ephemeris is read back, one line at a time, from the scratch files and output on the line printer. Finally, the date, time (MJD and fraction of a day) and components of position and velocity for the last integration step are printed. This information may be used to restart the integration at a future time. A sample line printer output is given in Fig.3.

7 DATA DECK

7.1 Overall description

This section describes the input data for POINT, which, for convenience, is treated here as a set of punched cards. There are 24 possible records which may appear in a data deck, and the number present for any one run will be not less than 4. The order of the cards is as follows:

(i) time card

(ii) control cards

(iii) station cards

(iv) orbit information cards

(v) covariance matrix.

Items (iii) and (v) are options and may be omitted in appropriate circumstances. All records are read in free format except for those specifying RAE or SDC elements and those specifying a covariance matrix. Listings of specimen data decks are given in Fig.4. (The first deck shown was used to generate the output given in Fig.3.) In the descriptions of the data cards which follow, all parameters and constants are referred to by their Fortran-variable names.

7.2 The time card

The time card contains five parameters specifying the epoch of the orbital elements, the time span of the integration, the interpolation interval and the time at which the first ephemerides are required.

MJD This specifies the MJD number of the epoch at which the orbital parameters of the satellite are defined.

EP This is the time of (the input) epoch, expressed as a fraction of a day relative to MJD.

Both MJD and EP are set to zero if the orbit is specified by SDC elements since the epoch is specified in the standard format (read by subroutine SDC2EL).

TIOR This specifies the time-interval of the ephemeris, in hours.

DT This is the required interpolation interval, in minutes (negative if the integration is backwards in time).

TINT This is the time at which the first ephemerides are required in hours from epoch. If TINT = 0.0 , it is assumed to be DT minutes from epoch.

7.3 First control card

This control card contains eight parameters (all integer), which control the working of the program.

I This indicates the form in which the orbital parameters are provided (see section 7.6). If I = 1 , a set of injection conditions in kilometres, kilometres per second and degrees is read in the order

specified in section 2. If I = 2 , a set of position and velocity components in kilometres, kilometres per second is read. If I = 3 a set of osculating elements in kilometres and degrees is provided. If I = 4 , SDC two-line elements are read. If I = 5 , RAE mean elements are provided.

NM — This indicates the matrix operations to be performed (see section 7.7). If NM = 0 , there are no matrix operations. If NM = 1 , a covariance matrix in terms of injection conditions (with units feet, feet per second, degrees) is read, and a matrix in terms of position and velocity components (in units kilometres, kilometres per second) derived from it. If NM = 2 , a covariance matrix in terms of position and velocity components in kilometres, kilometres per second is read.

NOTPT — If NOTPT > 0 , an ephemeris of time and corresponding position and velocity components (in days, kilometres, kilometres per second) is written to an unformatted disc file at constant time intervals (DT).

NPV — If NPV = 1 , an ephemeris of time, position components, radial distance, velocity components and speed is written to the line printer.

NEL — If NEL = 1 , an ephemeris of time, osculating elements and radial distance is written to the line printer.

NAP — If NAP = 1 , a table of apses giving time, osculating elements and radial distance is written to the line printer.

NAE — If $1 \leqslant$ NAE $\leqslant 4$, tables of look angles for NAE stations giving times, azimuth, elevation, range and range rate are written to the line printer.

LSP — If LSP = 1 , luni-solar perturbations are included in the integration.

7.4 Second control card

This card contains one control parameter, one parameter which is used both as a control parameter and a data item and two data parameters.

C1 — Although the integration step length (H) is set in the BLOCK DATA segment, it may be modified to H/C1 by setting C1 $\neq$ 0 (C1 must be negative if backward integration is required).

ARMA This is the area-to-mass ratio of the satellite (m^2/kg). If set to 0.0, the air drag terms are excluded from the integration.

SMEAN This is the value of the solar 10.7cm radiation flux averaged over 3 days and measured in units of 10^{-22}W m^{-2} Hz^{-1}. If ARMA = 0.0 , then SMEAN = 0.0 .

SOLBAR This is the value of the solar 10.7cm radiation flux averaged over 3 months and measured in units of 10^{-22}W m^{-2} Hz^{-1}. If ARMA = 0.0 , then SOLBAR = 0.0 .

7.5 Station cards

Tables of look angles may be produced for NAE stations, where NAE $\leqslant 4$. One card is provided for each station and has four parameters.

SLT This is the station's geodetic latitude (degrees).

SLO This is the station's geocentric longitude (degrees).

R This is the station's height (m) above the geoid.

STANAM This is the station name of no more than 20 characters.

7.6 Orbit information cards

Orbit information may be supplied in one of five forms, each of which requires one or more cards. The form is indicated by the parameter I on the first control card.

I	Number of cards	Contents
1	1	Injection parameters (km, km/s, deg)
2	1	Geocentric position and velocity components (km, km/s)
3	1	Osculating elements (km, deg)
4	2	SDC two-line elements
5	5	RAE mean elements

7.7 Covariance matrix

The covariance matrix is supplied on twelve cards. The form is indicated by the parameter NM on the first control card. Each card contains three elements punched in 3E15.8 format. Cards one and two hold the first row, cards three and four the second row and so on.

8 DESCRIPTION OF OUTPUT

A specimen of the line printer output is shown in Fig.3. Time is given in days and fractions of a day, distances in kilometres and angles in degrees.

A total of six output options are available:

(1) An ephemeris of time, position components, radial distance, velocity components and speed written to the line printer.

(2) A propagated covariance matrix, printed after each interpolation.

(3) An ephemeris of time, position and velocity components written to an unformatted disc file.

(4) An ephemeris of time, osculating elements and radial distance written to the line printer.

(5) Tables of look angles for a maximum of four stations. Each table comprises time, azimuth, elevation, range and range rate; this information is only printed if the elevation is greater than or equal to 0.0°.

(6) A table of apses written to the line printer. Each entry gives the time, osculating elements and radial distance of the apse. If no apses are found, an appropriate comment is printed.

Finally at the end of each output, the time of the last integration step is printed together with the position and velocity components at that step.

All or any combination of the options may be chosen. However, option 2 is meant to be used with option 1 since the program is designed to print the propagated matrix after each line of option 1. If option 2 is chosen without option 1, the program will function correctly, but time will not be printed with each matrix.

9 PROGRAMS DERIVED FROM POINT

POINT has been designed as a versatile program with a number of input and output options. However, for some tasks which require much computation, these options may not be required and it may well be more efficient in terms of execution time and core store to produce a special version of the program. The on-line editor makes this a straightforward task.

Such a version was produced for the studies of the proposed Astronomical Roentgen Observatory (ASRO)[1]. Other versions which currently exist are POINT2, PTATMAN and PTDEC.

POINT2 is used to generate random transfer orbits for the synchronous mission analysis program SYNMAP and has been described elsewhere[12].

PTATMAN comprises the standard version of POINT plus the facility to include a gross attitude manoeuvre, such as would be carried out in a transfer orbit, to realign a spacecraft for the firing of an apogee boost motor. This manoeuvre is assumed to be impulsive (and applied at apogee) and the program is supplied with transverse, normal and radial delta-velocity components. An extra card, punched in free format, is inserted after the second control card in the standard POINT data deck. This card contains the apogee number (punched as an integer) and the three delta-velocity components. One other feature of this version is that the number of stations for which look angles may be produced has been increased from four to five, since this is the number usually associated with the launch phase of a synchronous orbit mission.

PTDEC is used to find the time at which a satellite decays, i.e. the time at which it falls below a given altitude. The program is provided with the area-mass ratio of the satellite and either a set of geocentric position and velocity components or a set of SDC two-line elements. Air drag, luni-solar perturbations and zonal harmonics up to J_9 are automatically included but tesseral harmonics are omitted. The orbit is integrated forward, and at each perigee, the osculating elements, perigee radius and preceding apogee radius are printed. The integration ceases when either the satellite's altitude falls below a given value or a specified time limit has elapsed.

Acknowledgment

The authors wish to thank Mr. A.W. Odell for his helpful advice and assistance with numerical techniques.

PRECEDING PAGE BLANK-NOT FILMED

Appendix A

POINT PROGRAM UNITS

ANGLE	reduces an angle to the range 0 to 2π
ANGL1	reduces an angle to the range $-\pi$ to π
BLOCK DATA	sets certain constants
COTOEL	converts coordinates to osculating elements
DEQRSPT	integrates differential equations
EAFKEP	determines eccentric anomaly from Kepler's equation
ELTOCO	converts osculating elements to coordinates
INFIND[(a)]	finds a named area on a disc file
INTFRC[(a)]	converts a number into its integer and fractional parts
INTPTA	computes the position and velocity components and covariance matrix by interpolation
INTPTB	computes the position and velocity components between integration steps by interpolation
IPTOCO	converts injection parameters to coordinates
MATMUL	performs matrix multiplication
MODAT	computes atmospheric density
ORBINT	controls orbit integration and interpolation
OUTPUT	transforms position and velocity into required form and stores
PDAUXP	auxiliary for DEQRSPT; computes perturbing accelerations
POSVEL	computes position and velocity components from RAE mean elements
PRINT	produces line printer output from information stored on discs
RAZEL	computes range, range rate, azimuth and elevation for given ground station
READEL	reads RAE 5-card orbital elements
REIP	reads input data
RLEASE[(b)]	releases peripheral unit
ROTATE	performs rotations about an axis through a given angle at a given angular velocity
SCPROD[(a)]	forms scalar product
SDC2EL	reads and converts SDC 2-line elements to RAE mean elements
SETPD	sets initial values of partial derivatives
SHPAUX	computes short-periodic perturbations
SMPOS	reads sun/moon coordinates from disc and interpolates
SOLVIN	performs interpolation
TRAMAT	transposes a matrix
TRINV	determines polar coordinates from cartesian (two-dimensional)
UTD4[(b)]	permits disc/core transfers

(a) The subprograms INFIND, INTFRC and SCPROD are written in PLAN and are used as semi-compiled segments.

(b) The subprograms RLEASE and UTD4, though not part of standard FORTRAN, are provided automatically by the 1900 series FORTRAN compilers, and are not described here.

Appendix B

SUBPROGRAM SPECIFICATIONS

FUNCTION ANGLE

Summary	- The function reduces an angle x (in radians) to the range $0 \leqslant x < 2\pi$.
Language	- 1900 Fortran.
Author	- Diana W. Scott (April 1969).
Function statement	- FUNCTION ANGLE(X).
Input argument	-
X	Angle x in radians.
Output function	-
ANGLE	- Value of $x \pm 2n\pi$, such that $0 \leqslant \mathrm{ANGLE} < 2\pi$.
Use of COMMON	- None.
Source deck	- 6 cards, including 1 comment card (ICL code).
Local storage used	- 1 real variable.
Subordinate subprograms	- None.

Explanation - The standard function AMOD is used to give the fractional part of $x/2\pi$. If this is negative, 2π is added to the result.

FUNCTION ANGL1

Summary	- The function reduces an angle x to the range $-\pi < x \leqslant \pi$.
Language	- 1900 Fortran.
Author	- Diana W. Scott (April 1969).
Function statement	- FUNCTION ANGL1(X)
Input argument	-
X	Angle x in radians.
Output function	-
ANGL1	The value of $x \pm 2n\pi$ such that $-\pi <$ ANGL1 $\leqslant \pi$.
Use of COMMON	- None.
Source deck	- 7 cards, including one common card (ICL code).
Local storage used	- 1 real variable.
Subordinate subprograms	- None.

Explanation - The standard function AMOD is used to give the fractional part of $x/2\pi$. If this is greater than π , 2π is subtracted. If it is less than, or equal to, $-\pi$, 2π is added.

BLOCK DATA (POINT)

Summary - The Fortran BLOCK DATA segment is used to set initial values for certain quantities stored in common blocks /PETURB/, /CINTEG/, /CONST/ and /CON/.

Language - 1900 Fortran.

Author - M.D. Palmer (January 1975).

Data in /PETURB/ -

Variable name	Value	Explanation
HALFCD	1.1	Product $\frac{1}{2}C_D$.
RATE	1.1	Ratio of angular velocity of the atmosphere to that of the earth.

Data in /CINTEG/ -

Variable name	Value	Explanation
H	$2\pi/96$	Integration step length.
PD (18)	All elements 0.0	Partial derivatives of position.
PDVEL (18)	All elements 0.0	Partial derivatives of velocity.

Data in /CONST/ -

Variable name	Value	Explanation
EMU	$398601.3\text{km}^3\text{s}^{-2}$	Earth's gravitational constant.
EJ2	1082.637E-6	Earth's second zonal harmonic J_2 .
EJ3	-2.5310E-6	Earth's third zonal harmonic J_3 .
EJ4	-1.6190E-6	Earth's fourth zonal harmonic J_4 .
C22	2.4369E-6	Tesseral harmonic coefficient C_{22} .
S22	-1.4005E-6	Tesseral harmonic coefficient S_{22} .
C33	0.7387E-6	Tesseral harmonic coefficient C_{33} .
S33	1.4343E-6	Tesseral harmonic coefficient S_{33} .
C44	-0.1846E-6	Tesseral harmonic coefficient C_{44} .
S44	0.2508E-6	Tesseral harmonic coefficient S_{44} .
C31	2.0192E-6	Tesseral harmonic coefficient C_{31} .
S31	0.2278E-6	Tesseral harmonic coefficient S_{31} .
C42	0.3444E-6	Tesseral harmonic coefficient C_{42} .
S42	0.7021E-6	Tesseral harmonic coefficient S_{42} .
ERAD	6378.163km	Earth's mean equatorial radius.

Variable name	Value	Explanation
EOMEGA	7.292115147E-5rad/s	Earth's angular velocity.
PREC	6.079E-12rad/s	Earth's precession rate.
SUNMU	1.327127E11km^3s^{-2}	Sun's gravitational constant.
SELMU	4902.756km^3s^{-2}	Moon's gravitational constant.

Data in /CON/ -

Variable name	Value	Explanation
EJ5	-0.246E-6	Earth's fifth zonal harmonic J_5.
EJ6	0.558E-6	Earth's sixth zonal harmonic J_6.
EJ7	-0.326E-6	Earth's seventh zonal harmonic J_7.
EJ8	-0.209E-6	Earth's eighth zonal harmonic J_8.
EJ9	-0.094E-6	Earth's ninth zonal harmonic J_9.
C43	1.0390E-6	Tesseral harmonic coefficient C_{43}.
S43	-0.1192E-6	Tesseral harmonic coefficient S_{43}.
C41	-0.5175E-6	Tesseral harmonic coefficient C_{41}.
S41	-0.4814E-6	Tesseral harmonic coefficient S_{41}.
C32	0.7783E-6	Tesseral harmonic coefficient C_{32}.
S32	-0.7552E-6	Tesseral harmonic coefficient S_{32}.

Source deck - 18 cards (ICL code).

SUBROUTINE COTOEL

Summary - The subroutine derives the standard elliptic (osculating) orbital elements of an earth satellite, given its position and velocity components.

Language - ASA Fortran (Standard Fortran 4).

Author - R.H. Gooding (May 1976).

Subroutine statement - SUBROUTINE COTOEL (X, Y, Z, XDOT, YDOT, ZDOT, EMU, A, E, ORBINC, RANODE, ARGPER, EM, EN).

Input arguments -

X, Y, Z — Geocentric cartesian components, (x,y,z), of the satellite, x towards the vernal equinox and z towards the north pole.

XDOT, YDOT, ZDOT — Velocity of the satellite, $(\dot{x},\dot{y},\dot{z})$, measured in the same coordinate system.

EMU — Earth's gravitational constant, μ .

Output arguments -

A — Semi-major axis, a , in the same units as x, y, z

E — Eccentricity, e .

ORBINC — Orbital inclination, i .

RANODE — Right ascension of the ascending node, Ω .

ARGPER — Argument of perigee, ω .

EM — Mean anomaly, M .

EN — Mean motion, n .

Use of COMMON - None.

Source deck - 30 cards, including four comment cards (ICL code).

Local storage used - 13 real variables.

Subordinate subprograms - The subroutine TRINV.

Explanation - Although the subroutine essentially transforms from the six components of the position and velocity of a satellite to its six orbital elements, a seventh input argument permits an arbitrary value of the constant μ

to be used; the seventh output argument, n, is derived from a and μ by the relation $n^2a^3 = \mu$ (Kepler's third law). This means that the subroutine may be used very generally; e.g. for a planet, taking the sun's μ and interpreting Ω as celestial longitude. Units of time and distance are arbitrary but must, of course, be consistent; all angles are in radians.

The subroutine has been written to give maximum accuracy. All angles are derived from knowledge of both sine and cosine, and in such an order that there is no difficulty near the singularities at $e = 0$, $i = 0$ and $i = \pi$. The general method is that of Brouwer and Clemence[13] (section 27 of chapter 1).

The first quantities to be derived are a, e and the eccentric anomaly, E. (NB The Fortran variable E refers to the eccentricity and not the eccentric anomaly.) These come from the relations

$$\frac{1}{a} = \frac{2}{r} - \frac{V^2}{\mu} ,$$

$$e \cos E = \frac{rV^2}{\mu} - 1$$

and

$$e \sin E = \frac{(x\dot{x} + y\dot{y} + z\dot{z})}{(\mu a)^{\frac{1}{2}}} ,$$

where $r = (x^2 + y^2 + z^2)^{\frac{1}{2}}$ and $V^2 = \dot{x}^2 + \dot{y}^2 + \dot{z}^2$.

Then M follows at once, since

$$M = E - e \sin E .$$

Note that if e is close to zero E is ill-determined, reflecting the indeterminancy of perigee - and in fact if $e = 0$, E is set to 0. This does not matter at all, since whatever position of perigee is specified by the value taken for E, the value of M is fully consistent with it.

The values of i, Ω and ω are derived from the basic matrix relation

$$\begin{pmatrix} \cos\omega & \sin\omega & 0 \\ -\sin\omega & \cos\omega & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos i & \sin i \\ 0 & -\sin i & \cos i \end{pmatrix} \begin{pmatrix} \cos\Omega & \sin\Omega & 0 \\ -\sin\Omega & \cos\Omega & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} P_x & P_y & P_z \\ Q_x & Q_y & Q_z \\ R_x & R_y & R_z \end{pmatrix} ,$$

where $P_x = (x/r)\cos E - \dot{x}(a/\mu)^{\frac{1}{2}} \sin E$,

$Q_x = (1 - e^2)^{-\frac{1}{2}}\left[(x/r)\sin E + \dot{x}(a/\mu)^{\frac{1}{2}}(\cos E - e)\right]$,

$R_x = (\mu a(1 - e^2))^{-\frac{1}{2}}(y\dot{z} - z\dot{y})$,

and similarly for P_y, P_z, Q_y, Q_z, R_y and R_z .

Thus i and Ω are derived from

$$\sin i \sin \Omega = R_x \ ,$$

$$-\sin i \cos \Omega = R_y \ ,$$

and

$$\cos i = R_z \ ,$$

with an indeterminacy in Ω when i is close to 0 or π (Ω is set to 0 if $\sin i = 0$).

Finally ω is determined from

$$\sin i \cos \omega = Q_z$$

and

$$\sin i \sin \omega = P_z$$

where we have to be sure that no difficulty arises over the e-singularity or either of the i-singularities.

There is no trouble near the e-singularity since the quantities $e \cos E$ and $e \sin E$ are used (through Q_z and P_z), in the derivation of ω , in the same ratio as in the derivation of E , so that $E + \omega$ is always correct. (When E is set conventionally to 0 because $e = 0$, $e \cos E$ is set to 1 (!) to ensure the correct ratio between Q_z and P_z .)

In the same way, ω has to be compatible with Ω near the i-singularity, and this is automatic when z and $\dot{z}$ are not both exactly zero, since Ω and ω both depend on the ratio of these two quantities. When z and $\dot{z}$ *are* both exactly zero, the correct value of ω is achieved by replacing P_z and Q_z by P_y and Q_y , with an additional factor to ensure that ω is in the correct quadrant.

SUBROUTINE DEQRSPT

Summary - The subroutine integrates a set of up to 22 simultaneous second-order differential equations of the form $\ddot{y}_i = f_i(t,y_1,\ldots,y_N, \dot{y}_1,\ldots,\dot{y}_N)$, $i = 1,\ldots,N$, using a Gaussian eighth-order second-sum predictor-corrector[14]. The integration is started using a Butcher sixth-order seven stage Runge-Kutta process[15] to set up the difference table required before the second-sum procedure can take over.

Language - 1900 Fortran.

Authors - A.W. Odell and G.J. Davison (April 1973).

Subroutine statement - SUBROUTINE DEQRSPT (NTRY, DAUX).

Input arguments -

NTRY 1 for an initialization entry to the subroutine, and 2 for all normal entries (see Explanation).

DAUX The auxiliary subroutine which evaluates $\ddot{y}_i$.

Output arguments - None.

Use of COMMON - Certain quantities in common block /CINTEG/ are used as follows:-

Input arguments in /CINTEG/ -

NEQ Number of equations (normally 4 or 22).

H Integration step size, h (positive or negative).

Input and output arguments in /CINTEG/ -

T Independent variable, t .

Y(22) Dependent variables, y_i .

YP(22) First derivatives, $\dot{y}_i$.

Y2P(22) Second derivatives, $\ddot{y}_i$, as computed by the auxiliary subroutine DAUX.

Output arguments in /CINTEG/ -

IAUX Set to -1 following the initialisation entry (NTRY=1), to 0 if T has been changed (during a normal entry), and to 1 otherwise.

Source deck - 218 cards (ICL code).

Local storage used - 10 integer variables, 1 logical variable, 63 real variables and 814 real-array elements.

Subordinate subprograms - The auxiliary subroutine, named in the call to DEQRSPT which takes the role of DAUX.

Explanation - DEQRSPT integrates a set of $N(\leqslant 22)$ simultaneous second-order differential equations of the form $\ddot{y}_i = f_i(t,y_1,\ldots y_N,\dot{y}_1,\ldots\dot{y}_N)$, $i = 1,\ldots,N$, using a Gaussian eighth-order second-sum predictor corrector. The integration is started using a Butcher (6,7) R-K process to set up the difference table required before the second-sum procedure takes over.

Prior to any integration, DEQRSPT must first be called with NTRY = 1. This is the initialization entry in which the step size $h'(=\frac{1}{2}h)$ is set for the Butcher integration and DAUX is called to evaluate $\ddot{y}_i$ at t_0. All subsequent entries are made with NTRY = 2, and one integration step (two, whilst still in the Butcher mode) is performed before control is returned to the calling program.

If $y_{i,k}$ and $\dot{y}_{i,k}$ are the values of y_i and $\dot{y}_i$ at $t = t_k$ the formulae used in the next Butcher step to evaluate $y_{i,k+1}$, $\dot{y}_{i,k+1}$ and $\ddot{y}_{i,k+1}$, the values of y_i, $\dot{y}_i$ and $\ddot{y}_i$ at $t = t_k + h'$, are:

$$y_{i,k+1} = y_{i,k} + \sum_{s=1}^{7} w_s k_{is} ,$$

$$\dot{y}_{i,k+1} = \dot{y}_{i,k} + \sum_{s=1}^{7} w_s \ell_{is}$$

and

$$\ddot{y}_{i,k+1} = f_i(t_k + h', y_{1,k+1},\ldots y_{N,k+1}, \dot{y}_{1,k+1},\ldots \dot{y}_{N,k+1}) , \qquad i = 1,\ldots,N$$

where $$k_{is} = h' f_i\left(t_k + c_s h', y_{i,k} + \sum_{j=1}^{s-1} a_{sj}\ell_{ij}, \ell_{is}\right) ,$$

$$\ell_{is} = \dot{y}_{i,k} + \sum_{j=1}^{s-1} a_{sj}k_{ij} , \qquad s = 1,\ldots,N ,$$

and the remaining quantities are given by the following table

$$
\begin{array}{c|ccccccc}
0 & & & & & & & \\
c_2 & a_{21} & & & & & & \\
c_3 & a_{31} & a_{32} & & & & & \\
c_4 & a_{41} & a_{42} & a_{43} & & & & \\
c_5 & a_{51} & a_{52} & a_{53} & a_{54} & & & \\
c_6 & a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & & \\
c_7 & a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & \\
\hline
 & w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & w_7
\end{array}
=
\begin{array}{c|ccccccc}
0 & & & & & & & \\
\frac{1}{3} & \frac{1}{3} & & & & & & \\
\frac{2}{3} & 0 & \frac{2}{3} & & & & & \\
\frac{1}{3} & \frac{1}{12} & \frac{1}{3} & -\frac{1}{12} & & & & \\
\frac{1}{2} & -\frac{1}{16} & \frac{9}{8} & -\frac{3}{16} & -\frac{3}{8} & & & \\
\frac{1}{2} & 0 & \frac{9}{8} & -\frac{3}{8} & -\frac{3}{4} & \frac{1}{2} & & \\
1 & \frac{9}{44} & -\frac{9}{11} & \frac{63}{44} & \frac{18}{11} & 0 & -\frac{16}{11} & \\
\hline
 & \frac{11}{120} & 0 & \frac{27}{40} & \frac{27}{40} & -\frac{4}{15} & -\frac{4}{15} & \frac{11}{120}
\end{array}
$$

Given $y_{i,0}$ and $\dot{y}_{i,0}$, $\ddot{y}_{i,0}$ is obtained from DAUX, and the first two Butcher steps of size h' then give $y_{i,1}$, $\dot{y}_{i,1}$ and $\ddot{y}_{i,1}$ at time $t_1 = t_0 + h$. The process is repeated until $y_{i,8}$, $\dot{y}_{i,8}$ and $\ddot{y}_{i,8}$ are obtained after a total of 16 steps (eight calls to the subroutine). The Butcher process is now complete.

On the ninth call (with NTRY = 2) to DEQRSPT the following difference table is constructed for each $\ddot{y}_i \equiv f_i$.

$$
\begin{array}{ccccccccccc}
 & & \ddot{y}_{i,0} & & & & & & & & \\
 & & & \nabla^{1}_{i,1} & & & & & & & \\
 & & \ddot{y}_{i,1} & & \nabla^{2}_{i,2} & & & & & & \\
 & & & \nabla^{1}_{i,2} & & \nabla^{3}_{i,3} & & & & & \\
 & & \ddot{y}_{i,2} & & \nabla^{2}_{i,3} & & \nabla^{4}_{i,4} & & & & \\
 & & & \nabla^{1}_{i,3} & & \nabla^{3}_{i,4} & & \nabla^{5}_{i,5} & & & \\
 & & \ddot{y}_{i,3} & & \nabla^{2}_{i,4} & & \nabla^{4}_{i,5} & & \nabla^{6}_{i,6} & & \\
 & & & \nabla^{1}_{i,4} & & \nabla^{3}_{i,5} & & \nabla^{5}_{i,6} & & \nabla^{7}_{i,7} & \\
\nabla^{-2}_{i,3} & & \ddot{y}_{i,4} & & \nabla^{2}_{i,5} & & \nabla^{4}_{i,6} & & \nabla^{6}_{i,7} & & \nabla^{8}_{i,8} \\
 & \nabla^{-1}_{i,4} & & \nabla^{1}_{i,5} & & \nabla^{3}_{i,6} & & \nabla^{5}_{i,7} & & \nabla^{7}_{i,8} & \\
\nabla^{-2}_{i,4} & & \ddot{y}_{i,5} & & \nabla^{2}_{i,6} & & \nabla^{4}_{i,7} & & \nabla^{6}_{i,8} & & \nabla^{8}_{i,9} \\
 & \nabla^{-1}_{i,5} & & \nabla^{1}_{i,6} & & \nabla^{3}_{i,7} & & \nabla^{5}_{i,8} & & \nabla^{7}_{i,9} & \\
\nabla^{-2}_{i,5} & & \ddot{y}_{i,6} & & \nabla^{2}_{i,7} & & \nabla^{4}_{i,8} & & \nabla^{6}_{i,9} & & \\
 & \nabla^{-1}_{i,6} & & \nabla^{1}_{i,7} & & \nabla^{3}_{i,8} & & \nabla^{5}_{i,9} & & & \\
\nabla^{-2}_{i,6} & & \ddot{y}_{i,7} & & \nabla^{2}_{i,8} & & \nabla^{4}_{i,9} & & & & \\
 & \nabla^{-1}_{i,7} & & \nabla^{1}_{i,8} & & \nabla^{3}_{i,9} & & & & & \\
\nabla^{-2}_{i,7} & & \ddot{y}_{i,8} & & \nabla^{2}_{i,9} & & & & & & \\
 & \nabla^{-1}_{i,8} & & \nabla^{1}_{i,9} & & & & & & & \\
\nabla^{-2}_{i,8} & & \ddot{y}_{i,9} & & & & & & & & \\
 & \nabla^{-1}_{i,9} & & & & & & & & & \\
\nabla^{-2}_{i,9} & & & & & & & & & &
\end{array}
$$

First the known $\ddot{y}_i$ are differenced to give the part of the table to the right of the $\ddot{y}$, and then the first and second sums $\nabla^{-1}_{i,4}$ and $\nabla^{-2}_{i,3}$ are formed, using the equations

$$\nabla^{-2}_{i,3} = h^{-2}y_{i,4} - B_0\ddot{y}_{i,4} - B_2\nabla^{2}_{i,5} - B_4\nabla^{4}_{i,6} - B_6\nabla^{6}_{i,7} - B_8\nabla^{8}_{i,8}$$

and

$$\nabla^{-1}_{i,4} = h^{-1}\dot{y}_{i,4} - A_0\ddot{y}_{i,4} - A_1\nabla^1_{i,5} - A_2\nabla^2_{i,5} - A_3\nabla^3_{i,6} - A_4\nabla^4_{i,6} - A_5\nabla^5_{i,7} - A_6\nabla^6_{i,7} - A_7\nabla^7_{i,8} - A_8\nabla^8_{i,8} \ .$$

The remaining ∇^{-2} and ∇^{-1} quantities above the dotted line are defined on the basis that the difference between any entry and the entry above must equal the entry on the right. Actually only $\nabla^{-1}_{i,8}$ and $\nabla^{-2}_{i,8}$ are required explicitly and these are given by

$$\nabla^{-1}_{i,8} = \nabla^{-1}_{i,4} + \ddot{y}_{i,5} + \ddot{y}_{i,6} + \ddot{y}_{i,7} + \ddot{y}_{i,8}$$

and

$$\nabla^{-2}_{i,8} = \nabla^{-2}_{i,3} + 5\nabla^{-1}_{i,4} + 4\ddot{y}_{i,5} + 3\ddot{y}_{i,6} + 2\ddot{y}_{i,7} + \ddot{y}_{i,8} \ .$$

A table containing the coefficients used in the above and following equations is appended.

The ninth normal call to the subroutine continues with an integration step, using the Gaussian (predictor) formulae:-

$$y_{i,9} = h^2\left(\nabla^{-2}_{i,8} + C_0\ddot{y}_{i,8} + C_1\nabla^1_{i,8} + C_2\nabla^2_{i,8} + C_3\nabla^3_{i,8} + C_4\nabla^4_{i,8} + C_5\nabla^5_{i,8} + C_6\nabla^6_{i,8} + C_7\nabla^7_{i,8} + C_8\nabla^8_{i,8}\right)$$

and

$$\dot{y}_{i,9} = h\left(\nabla^{-1}_{i,8} + F_0\ddot{y}_{i,8} + F_1\nabla^1_{i,8} + F_2\nabla^2_{i,8} + F_3\nabla^3_{i,8} + F_4\nabla^4_{i,8} + F_5\nabla^5_{i,8} + F_6\nabla^6_{i,8} + F_7\nabla^7_{i,8} + F^8_{i,8}\right) \ ,$$

which require only the quantities immediately above the dotted line in the table.

$\ddot{y}_{i,9}$ is then obtained using DAUX and the row of differences under the diagonal line found. This row is then used to obtain corrected values of $y_{i,9}$ and $\dot{y}_{i,9}$ using the equations

$$y_{i,9} = h^2\Big(\nabla^{-2}_{i,8} + D_0\ddot{y}_{i,9} + D_1\nabla^1_{i,9} + D_2\nabla^2_{i,9} + D_3\nabla^3_{i,9} + D_4\nabla^4_{i,9}$$
$$+ D_5\nabla^5_{i,9} + D_6\nabla^6_{i,9} + D_7\nabla^7_{i,9} + D_8\nabla^8_{i,9}\Big)$$

and

$$\dot{y}_{i,9} = h\Big(\nabla^{-1}_{i,8} + E_0\ddot{y}_{i,9} + E_1\nabla^1_{i,9} + E_2\nabla^2_{i,9} + E_3\nabla^3_{i,9} + E_4\nabla^4_{i,9}$$
$$+ E_5\nabla^5_{i,9} + E_6\nabla^6_{i,9} + E_7\nabla^7_{i,9} + E_8\nabla^8_{i,9}\Big) .$$

$\ddot{y}_{i,9}$ is then redetermined, and the row of differences out to $\nabla^8_{i,9}$ under the dotted line recalculated. This row is then used to obtain the final corrected values of $y_{i,9}$ and $\dot{y}_{i,9}$ using the above equations.

Coefficients for A_i, B_i, C_i, D_i, E_i and F_i

i	0	1	2	3	4	5	6	7	8
A_i	$-\frac{1}{2}$	$-\frac{1}{12}$	$\frac{1}{24}$	$\frac{11}{720}$	$-\frac{11}{1440}$	$-\frac{191}{60480}$	$\frac{191}{120960}$	$\frac{2497}{3628800}$	$-\frac{2497}{7257600}$
B_i	$\frac{1}{12}$	0	$-\frac{1}{240}$	0	$\frac{31}{60480}$	0	$-\frac{289}{3628800}$	0	$\frac{317}{22809600}$
C_i	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{19}{240}$	$\frac{3}{40}$	$\frac{863}{12096}$	$\frac{275}{4032}$	$\frac{33953}{518400}$	$\frac{8183}{129600}$	$\frac{3250433}{53222400}$
D_i	$\frac{1}{12}$	0	$-\frac{1}{240}$	$-\frac{1}{240}$	$-\frac{221}{60480}$	$-\frac{19}{6048}$	$-\frac{9829}{3628800}$	$-\frac{407}{172800}$	$-\frac{330157}{159667200}$
E_i	$-\frac{1}{2}$	$-\frac{1}{12}$	$-\frac{1}{24}$	$-\frac{19}{720}$	$-\frac{3}{160}$	$-\frac{863}{60480}$	$-\frac{275}{24192}$	$-\frac{33953}{3628800}$	$-\frac{8183}{1036800}$
F_i	$\frac{1}{2}$	$\frac{5}{12}$	$\frac{3}{8}$	$\frac{251}{720}$	$\frac{95}{288}$	$\frac{19087}{60480}$	$\frac{5257}{17280}$	$\frac{1070017}{3628800}$	$\frac{25713}{89600}$

FUNCTION EAFKEP

Summary - The function solves Kepler's equation; i.e. it provides the eccentric anomaly of a celestial body, E, given the orbital eccentricity, e, and mean anomaly, M. Kepler's equation is $M = E - e \sin E$.

Language - ASA Fortran (Standard Fortran 4).

Author - R.H. Gooding (May 1976).

Function statement - FUNCTION EAFKEP (EM, ECC).

Input arguments -

EM Mean anomaly, M (radians).

ECC Eccentricity, e.

Output function -

EAFKEP Eccentric anomaly, E.

Use of COMMON - None.

Source deck - 13 cards, including 3 comment cards (ICL code).

Local storage used - 3 real variables.

Subordinate subprograms - None.

Explanation - A first approximation to E is given by $E_1 = M$. Improved approximations are given by Newton's method; thus

$$E_{i+1} = E_i + (M - E_i + e \sin E_i)/(1 - e \cos E_i) .$$

The process ends after three iterations if $e < 0.003$, and otherwise after five. This ensures sufficient accuracy at all times, while making the subroutine independent of the word-length of the computer. (If the magnitude of $|E_{i+1} - E_i|$ were used as a criterion for convergence, the numerical value it was compared with would have to vary from computer to computer.)

SUBROUTINE ELTOCO

Summary - The subroutine converts osculating Kepler elements into position and velocity components.

Language - ASA Fortran (Standard Fortran 4).

Author - A.W. Odell (August 1970).

Subroutine statement - SUBROUTINE ELTOCO (A, E, EI, BO, SO, EM, EMU, X, Y, Z, XDOT, YDOT, ZDOT).

Input arguments -

A	Semi-major axis, a .
E	Eccentricity, e .
EI	Orbital inclination, i .
BO	Right ascension of ascending node, Ω .
SO	Argument of perigee, ω .
EM	Mean anomaly M if EMU ≥ 0 , and true anomaly v if EMU < 0 .
EMU	Earth's gravitational constant, μ , the absolute value is used.

Output arguments -

X,Y,Z	The geocentric cartesian position coordinates (x,y,z) of the satellite, x towards the vernal equinox and z towards the north pole.
XDOT,YDOT,ZDOT	Velocity coordinates $(\dot{x},\dot{y},\dot{z})$ of the satellite measured in the same coordinate system.

Use of COMMON - None.

Source deck - 26 cards (ICL code).

Local storage used - 7 real variables.

Subordinate subprograms - The function EAFKEP and the subroutine ROTATE.

Explanation - If M is supplied, it is converted to the eccentric anomaly E by the function EAFKEP, and $\sin v$ and $\cos v$ are computed using:

$$\sin v = (1 - e^2)^{\frac{1}{2}} \sin E \Big/ (1 - e \cos E)$$

and

$$\cos v = (\cos e - e) \Big/ (1 - e \cos E) \quad .$$

Given $\sin v$ and $\cos v$, $r, x, y, z, \dot{x}, \dot{y}, \dot{z}$ are computed using:

$$r = a(1 - e^2) \Big/ (1 + e \cos v)$$

$$x = r \cos v \quad ,$$

$$y = r \sin v \quad ,$$

$$z = 0 \quad ,$$

$$\dot{x} = - \left[\mu/(a(1 - e^2))\right]^{\frac{1}{2}} \sin v \quad ,$$

$$\dot{y} = \left[\mu/(a(1 - e^2))\right]^{\frac{1}{2}} (e + \cos v)$$

and

$$\dot{z} = 0 \quad .$$

Three rotations are then performed using subroutine ROTATE, one about the normal to the orbital plane (z-axis) by an angle ω , one about the nodal line (x-axis) by an angle i and one about the polar axis (z-axis) by an angle Ω , giving the required result.

If M is supplied then e should be <1. If v is supplied this limitation does not apply.

FUNCTION INFIND

<u>Summary</u> - The function finds the location of a named area on a specified disc file.

<u>Language</u> - PLAN for use with 1900 Fortran.

<u>Author</u> - A.W. Odell (July 1973).

<u>Function statement</u> - FUNCTION INFIND (NAME).

<u>Input argument</u> -

NAME — Name of an area on the disc file up to 12 characters in length; must be array element or text (Hollerith) constant.

<u>Output function</u>

INFIND — Number specifying an area on the disc file, 0 if the name is not found in the index.

<u>Use of COMMON</u> - The first 128 integer locations of blank common are used as temporary working space.

<u>Source deck</u> - 37 cards, including 3 comment cards (IBM 'bcd' code).

<u>Local storage used</u> - 37 words for program, 7 words for data.

<u>Subordinate subprograms</u> - The subroutine UTD4.

<u>Explanation</u> - The subroutine assumes that an index has been set up on the disc file using subroutine INITD and that information has been put in the index using subroutine ADDINF. Associated with each name in the index is a number specifying an area on the disc file. If this name is not found, INFIND is set to 0; otherwise it is set to the associated number.

<u>Remark</u>: A Fortran version of this subroutine exists.

FUNCTION INTFRC

Summary - The function computes the integral and fractional parts of a number.

Language - PLAN, for use with 1900 Fortran.

Author - A.W. Odell (February 1971).

Function statement - FUNCTION INTFRC (X).

Input and output arguments -

X Real number, which is truncated to its fractional part.

Output function -

INTFRC Integral part of X.

Use of COMMON - None.

Source deck - 23 cards including 3 comment cards (ICL code).

Local storage used - 18 words for program.

Subordinate subprograms - None.

Explanation - X is split into its mathematical integral part and its fractional part. For example:

X = - 3.4

I = INTFRC (X)

would result in I being set to -4 and X to 0.6.

A Fortran version of this subroutine is also available.

SUBROUTINE INTPTA

Summary - The subroutine calculates the cartesian components of the geocentric position and velocity of an earth satellite at constant time intervals by interpolation between two given points in the orbit. These points are defined by position, velocity and acceleration components at specified times. If required, the subroutine will also interpolate for the elements of the covariance matrix.

Language - 1900 Fortran.

Author - M.D. Palmer (July 1974)

Subroutine statement - SUBROUTINE INTPTA (TP, TIM, PPVA, PD1, PDV, T2, ABSTIM).

Input arguments -

TP — Time (t_1) prior to the first interpolation, measured in days from epoch.

TIM — Time (t_2) after the first interpolation, measured in days from epoch.

(When integrating backwards, TP and TIM will both be negative.)

PPVA(9) — Satellite geocentric position, velocity and acceleration components at time t_1
$(x_1, y_1, z_1, \dot{x}_1, \dot{y}_1, \dot{z}_1, \ddot{x}_1, \ddot{y}_1, \ddot{z}_1) \equiv (\underline{r}_1, \underline{v}_1, \dot{\underline{v}}_1)$.

PD1(18) — Partial derivatives of position at time t_1 ,
$\partial \underline{r}_1 / \partial (\underline{r}_0, \underline{v}_0)$

PDV(18) — Partial derivatives of velocity at time t_1 ,
$\partial \underline{v}_1 / \partial (\underline{r}_0, \underline{v}_0)$

PDA(18) — Partial derivatives of acceleration at time t_1 ,
$\partial \dot{\underline{v}}_1 / \partial (\underline{r}_0, \underline{v}_0)$.

ABSTIM — The absolute value of t_2 .

Output arguments -

T2 — The absolute value of the time of the first interpolation after t_2 .

Use of COMMON - Certain quantities in common blocks /CINTEG/ and /INTP/ are used as follows:

Input arguments in /CINTEG/ -

MJDOCH — Modified Julian day number of epoch.

N(1) — N(1) = 4 if only the position and velocity components are required and N(1) = 22 if the covariance matrix is also required.

PP(3) — Satellite's geocentric position components at time t_2 $\underline{r}_2 = (x_2, y_2, z_2)$.

PD(18) — Partial derivatives of position at time t_2 - $\partial \underline{r}_2 / \partial (\underline{r}_0, \underline{v}_0)$.

PDVEL(18) — Derivatives with respect to the independent variable s of the partial derivatives of position at t_2 - $\frac{d}{ds}\left[\frac{\partial \underline{r}_2}{\partial (\underline{r}_0, \underline{v}_0)}\right]$.

TVEL — Value of dt/ds at time t_2 .

PV(3) — Satellite's velocity components at time t_2, $\underline{v}_2 = (\dot{x}_2, \dot{y}_2, \dot{z}_2)$.

PA(3) — Satellite's acceleration components at time t_2 , $\underline{\dot{v}}_2 = (\ddot{x}_2, \ddot{y}_2, \ddot{z}_2)$.

PDACCT(18) — Partial derivatives of acceleration at time t_2, $\partial \underline{\dot{v}}_2 / \partial (\underline{r}_0, \underline{v}_0)$.

Input arguments in /INTP/ -

EP — Time of epoch, expressed as a fraction of a day relative to MJDOCH.

FDT — Interval between interpolations (days).

COVX — Covariance matrix in terms of the geocentric position and velocity components at epoch, $\mathrm{cov}(\underline{r}_0, \underline{v}_0)$.

NOTPT — If NOTPT > 0 , the interpolated output is stored in a disc file.

Input and output arguments in /INTP/ -

TINT — On input, the time (t) of the first interpolation measured in days from epoch. On output, the time of the first interpolation after t_2 , in days from epoch. (If integrating backwards, TINT is negative.)

Subordinate subprograms - The subroutines COTOEL, MATMUL, OUTPUT, RAZEL, TRAMAT and TRINV and the functions ANGLE and INTFRC.

Local storage used - 108 real array elements, 24 real variables and 3 integer variables.

Source deck - 68 cards (ICL code).

Explanation - The subroutine is designed to carry out a series of interpolations at constant time intervals between consecutive steps in an orbit integration. After the interpolated position and velocity components have been obtained, OUTPUT is called to execute the output options available with POINT. If NOTPT > 0 , the MJD, fraction of a day, position and velocity components are stored in an unformatted disc file (channel 5).

The formulae used for interpolation are:

$$f(t) = q^3\left[(1 + 3p + 6p^2)f(t_1) + hp(1 + 3p)f'(t_1) + \frac{h^2}{2}p^2f''(t_1)\right] + p^3\left[(1 + 3q + 6q^2)f(t_2) - hq(1 + 3q)f'(t_2) + \frac{h^2}{2}q^2f''(t_2)\right]$$

for position components, and

$$f'(t) = 30h^{-1}p^2q^2[f(t_2) - f(t_1)] + q^2(1 + 5p)(1 - 3p)f'(t_1) + p^2(1 + 5q)(1 - 3q)f'(t_2) + \frac{h}{2}pq^2(2 - 5p)f''(t_1) - \frac{h}{2}p^2q(2 - 5q)f''(t_2)$$

for velocity components, where $h = t_2 - t_1$, $p = (t - t_1)/h$, $q = 1 - p$ and $f(t_i)$, $f'(t_i)$, $f''(t_i)$ are the position, velocity and acceleration components at time t_i .

The subroutine will, if required, interpolate for the partial derivatives of position and velocity, and use them to obtain a covariance matrix, $\underline{X}$, at the required time by means of the transformation

$$\underline{X} = \underline{A}\ \mathrm{cov}(\underline{r_0},\underline{v_0})\ \underline{A}^T$$

where $\underline{A} = \begin{bmatrix} \frac{\partial x}{\partial x_0} & \frac{\partial x}{\partial y_0} & \frac{\partial x}{\partial z_0} & \frac{\partial x}{\partial \dot{x}_0} & \frac{\partial x}{\partial \dot{y}_0} & \frac{\partial x}{\partial \dot{z}_0} \\ \frac{\partial y}{\partial x_0} & \frac{\partial y}{\partial y_0} & \frac{\partial y}{\partial z_0} & \frac{\partial y}{\partial \dot{x}_0} & \frac{\partial y}{\partial \dot{y}_0} & \frac{\partial y}{\partial \dot{z}_0} \\ \frac{\partial z}{\partial x_0} & \frac{\partial z}{\partial y_0} & \frac{\partial z}{\partial z_0} & \frac{\partial z}{\partial \dot{x}_0} & \frac{\partial z}{\partial \dot{y}_0} & \frac{\partial z}{\partial \dot{z}_0} \\ \frac{\partial \dot{x}}{\partial x_0} & \frac{\partial \dot{x}}{\partial y_0} & \frac{\partial \dot{x}}{\partial z_0} & \frac{\partial \dot{x}}{\partial \dot{x}_0} & \frac{\partial \dot{x}}{\partial \dot{y}_0} & \frac{\partial \dot{x}}{\partial \dot{z}_0} \\ \frac{\partial \dot{y}}{\partial x_0} & \frac{\partial \dot{y}}{\partial y_0} & \frac{\partial \dot{y}}{\partial z_0} & \frac{\partial \dot{y}}{\partial \dot{x}_0} & \frac{\partial \dot{y}}{\partial \dot{y}_0} & \frac{\partial \dot{y}}{\partial \dot{z}_0} \\ \frac{\partial \dot{z}}{\partial x_0} & \frac{\partial \dot{z}}{\partial y_0} & \frac{\partial \dot{z}}{\partial z_0} & \frac{\partial \dot{z}}{\partial \dot{x}_0} & \frac{\partial \dot{z}}{\partial \dot{y}_0} & \frac{\partial \dot{z}}{\partial \dot{z}_0} \end{bmatrix}$

and $\mathrm{cov}(r_0,v_0)$ is the covariance matrix, in terms of the radial distance $\underline{r_0}$, and satellite velocity, $\underline{v_0}$, at the start of the integration.

$\underline{A}$ is obtained by interpolation between the partial derivatives of position and velocity at t_1 and t_2 , the computation requiring the partial derivatives of acceleration at those times. The partial derivatives of position are held in the arrays PD1 and PD respectively, the order of storage being

PD(1)-PD(6) $\partial x/\partial x_0$, $\partial x/\partial y_0$, $\partial x/\partial z_0$, $\partial x/\partial \dot{x}_0$, $\partial x/\partial \dot{y}_0$, $\partial x/\partial \dot{z}_0$

PD(7)-PD(12) $\partial y/\partial x_0$, $\partial y/\partial y_0$, $\partial y/\partial z_0$, $\partial y/\partial \dot{x}_0$, $\partial y/\partial \dot{y}_0$, $\partial y/\partial \dot{z}_0$

PD(13)-PD(18) $\partial z/\partial x_0$, $\partial z/\partial y_0$, $\partial z/\partial z_0$, $\partial z/\partial \dot{x}_0$, $\partial z/\partial \dot{y}_0$, $\partial z/\partial \dot{z}_0$

and similarly for PD(1).

The partial derivatives of velocity at t_1 are held in the array PDV, the order of storage being,

PDV(1)-PDV(6) $\partial \dot{x}/\partial x_0$, etc.

PDV(7)-PDV(12) $\partial \dot{y}/\partial x_0$, etc.

PDV(13)-PD(18) $\partial \dot{z}/\partial x_0$, etc.

The partial derivatives of velocity at t_2 are not available explicitly but are found from

$$\frac{d}{ds}\left[\frac{\partial \underline{\underline{r}}_2}{\partial(\underline{r}_0,\underline{v}_0)}\right] \Bigg/ \frac{dt}{ds}$$

where s is the independent variable defined by $dt/ds = r^{3/2}\mu^{-\frac{1}{2}}$. dt/ds is the Fortran variable TVEL and

$$\frac{d}{ds}\left[\frac{\partial \underline{\underline{r}}_2}{\partial(\underline{r}_0,\underline{v}_0)}\right]$$

is the array PDVEL.

The partial derivatives of acceleration at t_1 and t_2 are held in the arrays PDA and PDACCT respectively, the order of storage being

PDA(1)-PDA(6) $\partial \ddot{x}/\partial x_0$, etc.

PDA(7)-PDA(12) $\partial \ddot{y}/\partial x_0$, etc.

PDA(13)-PDA(18) $\partial \ddot{z}/\partial x_0$, etc.

and similarly for PDACCT.

The matrix X is written to the line printer and, if NOTPT > 0, it is also written to the unformatted disc file.

SUBROUTINE INTPTB

Summary - The subroutine calculates the cartesian components of the geocentric position and velocity of an earth satellite at a given time by interpolation between two points in the orbit. These points are defined by position, velocity and acceleration components at specified times.

Language - 1900 Fortran.

Author - M.D. Palmer (July 1974)

Subroutine statement - SUBROUTINE INTPTB (TINT, TP, TIMET, PPVA, PT, VT).

Input arguments -

TINT	Time, t, at which the interpolation is required.
TP	Time, t_1, of the point prior to the interpolation.
TIMET	Time, t_2, of the point after the interpolation.
PPVA(9)	Satellite's geocentric position, velocity and acceleration components at time t_1 (x_1, y_1, z_1 etc.).

Output arguments -

PT(3)	Satellite's geocentric position components at time t (x,y,z).
VT(3)	Satellite's geocentric velocity components at time t $(\dot{x},\dot{y},\dot{z})$.

Use of COMMON - Certain arguments in the common block /CINTEG/ are used as follows:

Input arguments -

PP(3)	Satellite's geocentric position components at time t_2 (x_2,y_2,z_2).
V(3)	Satellite's geocentric velocity components at time t_2 $(\dot{x}_2,\dot{y}_2,\dot{z}_2)$.
XACCT(3)	Satellite's geocentric acceleration components at time t_2 $(\ddot{x}_2,\ddot{y}_2,\ddot{z}_2)$.

Local storage used - 23 real variables and 1 integer variable.

Subordinate subprograms - None.

Source deck - 40 cards (ICL code).

Explanation - The subroutine obtains a satellite's geocentric position and velocity components at a given time by interpolation between two points in the orbit. These will normally be adjacent steps in a numerical integration.

These equations used for interpolation are:

$$f(t) = q^3\left[(1 + 3p + 6p^2)f(t_1) + hp(1 + 3p)f'(t_1) + \frac{h^2}{2}p^2f''(t_1)\right] + p^3\left[(1 + 3q + 6q^2)f(t_2) - hq(1 + 3q)f'(t_2) + \frac{h^2}{2}q^2f''(t_2)\right]$$

for position components and

$$f'(t) = 30h^{-1}p^2q^2\left[f(t_2) - f(t_1)\right] + q^2(1 + 5p)(1 - 3p)f'(t_1) + p^2(1 + 5q)(1 - 3q)f'(t_2) + \frac{h^2}{2}pq^2(2 - 5p)f''(t_1) - \frac{h}{2}p^2q(2 - 5q)f''(t_2)$$

for velocity components, where $h = t_2 - t_1$, $p = (t - t_1)/h$, $q = 1 - p$ and $f(t_i)$, $f'(t_i)$ and $f''(t_i)$ are the position, velocity and acceleration components at time t_i .

SUBROUTINE IPTOCO

Summary - The subroutine computes the geocentric cartesian components of the position and velocity of an earth satellite in PROP axes, given the date and time and a set of the standard parameters used to define orbit injection.

Language - USA Standard Fortran (USAS X3.9 - 1966).

Authors - G.E. Cook and R. Clarke (October 1972).

Subroutine statement - SUBROUTINE IPTOCO (X, Y, Z, XDOT, YDOT, ZDOT, V, CA, AZ, R, XLAT, XLONG, MJD, TIME).

Input arguments -

V	Speed, v .
CA	Climb angle, θ .
AZ	Azimuth of velocity vector, Ψ (measured E of N).
R	Radial distance, r .
XLAT	Geocentric latitude, ϕ .
XLONG	Longitude, λ .
MJD	Modified Julian Day number.
TIME	Time (fraction of a day) such that t is given by MJD + TIME .

Output arguments

X,Y,Z	Geocentric coordinates of satellite, x, y, z .
XDOT, YDOT, ZDOT	Geocentric components of velocity vector, $\dot{x}$, $\dot{y}$, $\dot{z}$.
XLONG	Longitude in inertial axes, λ' .

Use of COMMON - None.

Source deck - 22 cards (ICL code).

Local storage used - 8 real variables.

Subordinate subprograms - None.

Explanation - Units of length and time are arbitrary, except for the variables MJD and TIME; angles are in radians.

The components of position and velocity are evaluated relative to the following coordinate system: the origin 0 is at the earth's centre and Oz points towards the north pole; Ox lies in the plane of the true equator of date, but instead of pointing towards the true equinox of date it points towards a projection of the mean equinox of the epoch 1950.0 (MJD 33281.9234); Oy completes the right-handed system Oxyz.

The injection conditions are specified relative to an earth-fixed system with longitude measured eastwards from the Greenwich meridian. The longitude λ' relative to the inertial system defined above is found by adding $\hat{\theta}$ to λ, where $\hat{\theta}$ is the 'modified sidereal angle' given by

$$\hat{\theta} = 100.075542 + 360.985612288(t - 33282.0) \quad ,$$

i.e. the origin has been adjusted slightly from 1950.0. (The modified sidereal angle differs from sidereal time only because of the choice of a non-standard reference direction.)

The geocentric components of position and velocity are obtained from the following equations:

$$x = r \cos\phi \cos\lambda'$$

$$y = r \cos\phi \sin\lambda'$$

$$z = r \sin\phi$$

$$\dot{x} = v \left\{\sin\theta \cos\phi \cos\lambda' - \cos\theta \,(\cos\Psi \sin\phi \cos\lambda + \sin\Psi \sin\lambda)\right\}$$

$$\dot{y} = v \left\{\sin\theta \cos\phi \sin\lambda' + \cos\theta \,(\sin\Psi \cos\lambda - \cos\Psi \sin\phi \sin\lambda)\right\}$$

$$\dot{z} = v \left\{\sin\theta \sin\phi + \cos\theta \cos\Psi \cos\phi\right\} \quad .$$

SUBROUTINE MATMUL

Summary	- The subroutine performs matrix multiplications.
Language	- USA Standard Fortran (USAS X3.9 - 1966).
Author	- G.E. Cook (May 1970).
Subroutine statement	- SUBROUTINE MATMUL (EM1, EM2, EM3, II, KK, JJ, IA, IB).
Input arguments	-
EM1	Matrix of dimension (II,KK).
EM2	Matrix of dimension (KK,JJ).
II	Actual number of rows in EM1 .
KK	Actual number of columns in EM1 and rows in EM2 .
JJ	Actual number of columns in EM2 .
IA	Maximum number of rows in EM1 as given in the dimension statement of the calling segment.
IB	Maximum number of rows dimensioned for EM2 in the calling segment.
Output arguments	-
EM3	Matrix of dimension (II,JJ).
Use of COMMON	- None.
Source deck	- 13 cards, including 4 comment cards (ICL code).
Local storage used	- 3 integer variables.
Subordinate subprograms	- None.
Explanation	- The subroutine performs the matrix multiplication

$$M_1M_2 = M_3 \ .$$

SUBROUTINE MODAT

Summary - The subroutine evaluates upper-atmosphere density, density scale height and scale height gradient using a simple analytic model.

Language - 1900 Fortran.

Authors - G.E. Cook and K.J. Tomlinson (April 1969).

Subroutine statement - SUBROUTINE MODAT (HEIT, TINF, TGRADO, DEN, SCALHT, SHGRAD).

Input arguments -

HEIT	Height above the earth's surface, y .
TINF	Exospheric temperature, T_∞ .
TGRADO	Atmospheric temperature gradient dT/dy at the reference altitude, y_0 (120km).

Output arguments -

DEN	Atmospheric density, ρ (g/cm^3).
SCALHT	Density scale height (km).
SHGRAD	Density scale height gradient.

Use of COMMON - None.

Source deck - 46 cards, including 2 comment cards (ICL code).

Local storage used - 24 real array elements, 16 real variables, 1 integer variable.

Subordinate subprograms - None.

Explanation - The values of density and density scale height are obtained from a simple analytic model of the Earth's upper atmosphere. If g_0 denotes the local value of the acceleration due to gravity, the geopotential height above the reference altitude y_0 is defined by

$$\zeta = \int_{y_0}^{y} \{g(y)/g(y_0)\} dy \quad , \tag{1}$$

so that

$$\zeta = (y - y_0)(R + y_0)/(R + y) \quad , \tag{2}$$

R being the mean radius of the Earth. The temperature of the atmosphere is represented as a function of geopotential height by the expression

$$T(y) = T_\infty \{1 - a \exp(-\tau\zeta)\} \quad , \tag{3}$$

where T_∞ is the exospheric temperature and a and τ are constants defined by

$$a = 1 - \frac{T(y_0)}{T_\infty} \quad , \tag{4}$$

and

$$\tau = \frac{1}{T_\infty - T(y_0)} \left(\frac{dT}{dy}\right)_{y=y_0} \quad . \tag{5}$$

If the atmosphere is assumed to be in diffusive equilibrium above the reference altitude, the number density n_i of the ith constituent of molecular (or atomic) mass m_i is given by

$$\frac{1}{n_i}\frac{dn_i}{dy} = -\frac{m_i g}{kT} - \frac{1}{T}\frac{dT}{dy}(1 + \alpha) \quad , \tag{6}$$

where k is Boltzmann's constant and α is the thermal diffusion factor.

With the temperature profile (3), equation (6) can be integrated to give

$$n_i = n_i(y_0) \left\{\frac{1 - a}{1 - a \exp(-\tau\zeta)}\right\}^{1+\gamma_i+\alpha} \exp(-\gamma_i \tau\zeta) \quad ,$$

where $\gamma_i = \dfrac{m_i g(y_0)}{\tau k T_\infty}$.

The density ρ is given by

$$\rho = \sum_i n_i m_i \quad .$$

For the constant boundary conditions at the reference altitude of 120km we use the values assumed by Jacchia[16] in the construction of his static diffusion profiles:

$$T(120) = 355 \text{ K}$$

$$n(N_2) = 4.0 \times 10^{11} \text{ cm}^{-3}$$

$$n(O_2) = 7.5 \times 10^{10} \text{ cm}^{-3}$$

$$n(O) = 7.6 \times 10^{10} \text{ cm}^{-3}$$

$$n(He) = 3.4 \times 10^{7} \text{ cm}^{-3} \quad .$$

For hydrogen we also follow Jacchia and take the concentration at 500km to vary with T_∞ according to the relation

$$\log_{10} n(H;\, 500) = 73.13 - 39.40 \log_{10} T_\infty + 5.5 (\log_{10} T_\infty)^2 \quad .$$

The thermal diffusion factor α is taken as -0.4 for helium and as zero for the other constituents.

SUBROUTINE ORBINT

Summary - The subroutine starts and controls the integration of an elliptic orbit. During the integration, which may be forwards or backwards in time, it initiates interpolations at constant time intervals and, if required, will determine the apses of the orbit. The time of the apse, the osculating elements and radial distance are written to an unformatted disc file.

Language - 1900 Fortran.

Author - M.D. Palmer (February 1975).

Subroutine statement - SUBROUTINE ORBINT.

Use of COMMON - Certain quantities in the common blocks /CINTEG/, /CONST/, /CSMOON/ and /INTP/ are used as follows:

Input arguments in /CINTEG/ -

MJDOCH	Modified Julian day number of epoch.
N(1)	The number of equations to be integrated.
PP(3)	Satellite's geocentric position components, $\underline{r} = (x,y,z)$, initially at epoch $(\underline{r}_0)$ and subsequently at the latest integration step (MJDT + TIMET) (km).
PDV(18)	Partial derivatives of position at the latest integration step $\partial\underline{r}/\partial(\underline{r}_0,\underline{v}_0)$.
PDVEL(18)	Derivatives with respect to the independent variable s of the partial derivatives of position $$\frac{d}{ds}\left[\frac{\partial\underline{r}}{\partial(\underline{r}_0,\underline{v}_0)}\right] .$$
V(3)	Satellite's geocentric velocity components, $\underline{v} = (\dot{x},\dot{y},\dot{z})$, initially at epoch $(\underline{v}_0)$ and subsequently at the latest integration step (km/s).
A(3)	Satellite's geocentric acceleration components, $\dot{\underline{v}} = (\ddot{x},\ddot{y},\ddot{z})$ at the latest integration step (km/s^2).
PDACCT(18)	Partial derivatives of acceleration at the latest integration step $\partial\ddot{\underline{r}}/\partial(\underline{r}_0,\underline{v}_0)$.

Input arguments in /CONST/ -

EMU Earth's gravitational constant (km^3/s^2).

Input arguments in /CSMOON/ -

MJDT Modified Julian day number of the current time.

TIMET The current time, in fractions of a day, relative to MJDT.

Input arguments in /INTP/ -

EP The time of epoch measured in fractions of a day relative to MJDOCH.

TINT The time at which the next interpolation is required. Initially this is in hours from epoch and subsequently in days from epoch. If the integration is backwards in time, TINT must be negative.

TIOR Duration of the integration (hours).

Output arguments in /CINTEG/ -

XVEL, YVEL, ZVEL The initial values of dx/ds, dy/ds, dz/ds .

T The time, in seconds, of epoch relative to MJDOCH (i.e. $T = EP \times 86400.0$).

Input and output argument in /CINTEG/ -

TVEL The value of dt/ds at the latest integration step (s^{-1}).

Input and output arguments in /INTP/ -

DT On input, the interpolation interval in minutes. On output, the interpolation interval in fractions of a day. (If the integration is backwards in time, DT must be negative.)

NAP Flag controlling the detection of apses. Initially $NAP = 1$ if apses are to be found and subsequently $NAP = N + 1$ where N is the number of apses already found.

Local storage used - 69 real array variables, 17 real variables and 1 integer variable.

Subordinate subprograms - The subroutines COTOEL, DEQRSPT, INTPTA, INTPTB, MODAT, OUTPUT, POAUXP, SETPD, SMPOS and TRINV and the functions ANGLE, ANGL1, INTFRC, SOLVIN and SCPROD.

Source deck - 60 cards (ICL code).

Explanation - The subroutine starts and controls the forward, or backward, integration of a satellite orbit. The variables TVEL (dt/ds), XVEL (dx/ds), YVEL (dy/ds) and ZVEL (dz/ds) are set in terms of the independent variable s . If a covariance matrix is to be propagated, the subroutine SETPD is called to set initial values of the array PDVEL. T is set to the time of epoch in seconds, and TIOR and DT are converted to fractions of a day. TINT, the time of the first interpolation relative to epoch is input with units of hours and is converted into a time in days and fractions of a day.

The integration is started by calling DEQRSPT with the statement CALL DEQRSPT (IND, PDAUXP). On the first call IND = 1 and on the second and subsequent calls IND = 2 .

After each integration step, the time elapsed from epoch is calculated in days. If a table of apses is required, the length of the radius vector, r_i , and its rate of change $\dot{r}_i$, are determined. If $\dot{r}_i \dot{r}_{i-1} < 0$ there is an apse between the last two integration steps and SOLVIN is used to find its time and radial distance. INTPTB is called to find the position and velocity components at the apse and these are converted to osculating elements using COTOEL. This information is written unformatted to disc channel 4.

A check is made to see if the next required interpolation falls between the last two integration steps. If so, INPTA is called to carry out this interpolation and any others which fall between the two steps. When this is completed, INTPTB sets TINT to the time of the next required interpolation and returns control.

Before the next integration step, the geocentric position and velocity components, and if the matrix is to be propagated, the partial derivatives of position, velocity and acceleration are stored. The integration is continued until the time of the latest step, measured from epoch, exceeds TIOR.

SUBROUTINE OUTPUT

Summary - The subroutine, given a set of earth satellite geocentric position and velocity components (POSVEL), provides any combination of the following output options:

(i) Write POSVEL to line printer.

(ii) Convert POSVEL to osculating elements and write to disc.

(iii) Compute range, range rate, azimuth and elevation of the satellite from the specified ground stations and write to disc.

Language - 1900 Fortran.

Author - M.D. Palmer (February 1975).

Subroutine statement - SUBROUTINE OUTPUT (PT, VT, T, MJD, P).

Input arguments -

PT(3)	Satellite's geocentric position components (x,y,z) at epoch MJD + T (km).
VT(3)	Satellite's geocentric velocity components $(\dot{x},\dot{y},\dot{z})$ at epoch MJD + T (km).
MJD	Modified Julian day number of epoch.
T	Time, in fractions of a day relative to MJD.
P	Radial distance at epoch $(P = (x^2 + y^2 + z^2)^{\frac{1}{2}})$ (km).

Use of COMMON - Certain arguments in the common blocks /STATION/, /CONST/ and /INTP/ are used as follows:

Input argument in /CONST/ -

EMU	Earth's gravitational constant (km^3/s^2).

Input argument in /INTP/ -

NPV	If NPV > 0 , then MJD, T, POSVEL, radial distance and velocity are written to the line printer.

Input argument in /STATION/ -

IS	Number of ground stations.

Input and output arguments in /INTP/ -

NEL — On input, if NEL > 0 , the POSVEL is converted to osculating elements and written to disc. If this occurs NEL is incremented by 1.

NAE — On input, if NAE > 0 , the range, range rate, azimuth and elevation of the satellite relative to the specified ground stations are calculated and written to disc. If this occurs, NAE is incremented by 1.

Local storage used - 10 real variables, 1 integer variable.

Source deck - 29 cards (ICL code).

Subordinate subprograms - The subroutines COTOEL, RAZEL and TRINV and the function ANGLE.

Explanation - The subroutine is provided with an epoch and a corresponding POSVEL and radial distance. A number of output options are available and these are controlled by the flags NPV, NAE and NEL.

If NPV > 0 , the satellite's velocity is calculated and then MJD, T, x, y, z, radial distance, $\dot{x}$, $\dot{y}$, $\dot{z}$ and velocity are written to the line printer.

If NEL > 0 , the POSVEL is converted to osculating elements, using subroutine COTOEL. MJD, T , the osculating elements and radial distance are then written to an unformatted disc file on channel 3 and NEL incremented by 1.

If NAE > 0 , the sidereal angle at epoch is calculated and then look angles for each station are found using subroutine RAZEL. For each ground station, MJD, T, azimuth, elevation, range and range rate are written to the unformatted disc file on channel 3 and NAE incremented by 1.

SUBROUTINE PDAUXP

Summary - The subroutine calculates the cartesian components of the geocentric accelerations due to the gravitational attractions of the earth, sun and moon, atmospheric drag and the precession of the earth's polar axis. It will also, if required, evaluate the partial derivatives of the acceleration components with respect to the initial components of the position and velocity at epoch.

Language - 1900 Fortran.

Author - M.D. Palmer (November 1974).

Subroutine statement - Subroutine PDAUXP.

Use of COMMON - Certain quantities in the common blocks /CINTEG/, /CON/, /CONST/, /CSMOON/ and /PETURB/ are used as follows:

Input arguments in /CINTEG/ -

MJDOCH	Modified Julian day number of epoch.
NEQ	Number of equations being integrated (22 if partial derivatives of acceleration are required, 4 otherwise).
IAUX	Flag, see explanation.
X, Y, Z	Cartesian position components $(x,y,z) = \underline{r}$.
PD(18)	Partial derivatives of position with respect to the initial position $(\underline{r}_0)$ and velocity $(\underline{v}_0)$; $\partial\underline{r}/\partial(\underline{r}_0,\underline{v}_0)$.
XVEL, YVEL, ZVEL	The latest values of dx/ds, dy/ds and dz/ds .
T	Time, t , relative to MJDOCH (seconds).
PDVEL(18)	Derivatives with respect to the independent variable s of the partial derivatives of position

$$\frac{d}{ds}\left[\frac{\partial\underline{r}}{\partial(\underline{r}_0,\underline{v}_0)}\right] .$$

Input arguments in /CON/ -

EJ5, EJ6, EJ7, EJ8, EJ9	Coefficients of the earth's zonal harmonics, $J_5, J_6, \ldots, J_9$.
C32, S32 C41, S41 C43, S43	Fully normalized coefficients of certain tesseral harmonics, viz. $\overline{C_{32}}, \overline{S_{32}}, \overline{C_{41}}, \overline{S_{41}}, \overline{C_{43}}, \overline{S_{43}}$.

Input arguments in /CONST/ -

EMU	Earth's gravitational constant, μ_e .
EJ_2, EJ_3, EJ_4	Coefficients of the earth's zonal harmonics, J_2, J_3, J_4 .
C22, S22 C31, S31 C33, S33 C42, S42 C44, S44	Fully normalized coefficients of certain tesseral harmonics, viz. $\overline{C_{22}}, \overline{S_{22}}, \overline{C_{31}}, \overline{S_{31}}, \overline{C_{33}}, \overline{S_{33}}, \overline{C_{42}}, \overline{S_{42}}, \overline{C_{44}}, \overline{S_{44}}$.
ERAD	Mean equatorial radius of the earth, R .
EOMEGA	Mean rotation rate of the earth's polar axis, in rad/s.
SUNMU	Sun's gravitational constant, μ_s .
SELMU	Moon's gravitational constant, μ_m .

Input arguments in /CSMOON/ -

XS, YS, ZS	Cartesian components of the sun's position ($\underline{r}_s$) at the current time, computed if IAUX < 0 .
XM, YM, ZM	Cartesian components of the moon's position ($\underline{r}_m$) at the current time, computed if IAUX < 0 .

Input arguments in /PETURB/ -

SMEAN	Mean value over three days of the solar 10.7cm radiation flux in units of $10^{-22} W\ m^{-2}\ Hz^{-1}$.
SOLBAR	Mean value over three solar rotations of the 10.7cm radiation flux in units of $10^{-22} W\ m^{-2}\ Hz^{-1}$.
RATE	Angular velocity of the atmosphere, ω_a in rad/s.
ARMA	The product AM × HALFCD × 1000 where AM is the satellite's area-to-mass ratio and HALFCD is the quantity ½CD where CD is the satellite's drag coefficient. If ARMA = 0.0 , drag terms are excluded.

TNITE	Minimum night-time temperature (K).
LSP	If LSP > 0 , luni-solar perturbations are included.
NSR	If NSR > 0 , solar radiation pressure terms are included *but* see explanation.

Output arguments in /CINTEG/ -

XACC, YACC, ZACC	Values of d^2x/ds^2, d^2y/ds^2, d^2z/ds^2 at the current time.
TVEL	Value of dt/ds at the current time.
PDACC(18)	Second derivatives with respect to the independent variable s of the partial derivatives of position $$\frac{d^2}{ds^2}\left[\frac{\partial \underline{r}}{\partial(\underline{r}_0,\underline{v}_0)}\right] .$$
XVELT, YVELT, ZVELT	Cartesian components of velocity, $\dot{x}$, $\dot{y}$, $\dot{z}$, at the current time.
XACCT, YACCT, ZACCT	Cartesian components of acceleration, $\ddot{x}$, $\ddot{y}$, $\ddot{z}$, at the current time.
PDACCT(18)	Partial derivatives of acceleration with respect to the initial position and velocity at epoch $$\frac{d^2}{dt^2}\left[\frac{\partial \underline{r}}{\partial(\underline{r}_0,\underline{v}_0)}\right] .$$

Output arguments in /CSMOON/ -

MJDT	Modified Julian day number of the current time.
TIMET	The current time, in fractions of a day, relative to MJDT.

Source deck - 155 cards (ICL code).

Local storage used - 51 real variables and 3 integer variables.

Subordinate subprogram - The subroutines MODAT, SMPOS and TRINV and the functions ANGLE, ANGL1, INFIND and SCPROD.

Explanation - When the subroutine is called for the first time, or for a change of epoch, IAUX must be set to -1. For subsequent entries, when the time has changed since the previous entry, IAUX should be set to zero. If the

time has not changed IAUX may be set to 1 to prevent recomputation of certain terms. (All these cases are allowed for by the calling segment in POINT.)

The cartesian components of the geocentric acceleration acting on the satellite are found by adding contributions from the gravitational attractions of the earth, sun and moon, atmospheric drag and the precession of the earth's polar axis. The subroutine is structured so as to permit the easy insertion of solar radiation pressure at a future date.

The earth harmonic terms are:

$$\mu_e \sum_{n,m} R^n \Big/ r^{n+2} \mathcal{R}\left[E_{n,m}\left\{W^m\left(P_n^{(m+1)}\hat{\underline{z}} - P_{n+1}^{(m+1)}\hat{\underline{r}}\right) + mW^{m-1}e^{-j\theta}P_n^{(m)}(\hat{\underline{x}} + j\hat{\underline{y}})\right\}\right]$$

where the sidereal angle $\theta = \theta_0 + \omega_e t$,

$W \quad = (x + jy)/re^{j\theta})$,

$P_n^{(m)}$ is the mth derivative of the Legendre polynomial, $P_n(z/r)$

$E_{0,0} = 1$,

$E_{n,0} = - J_n$

and

$$E_{n,m} = \left[2(2n + 1)(n - m)!\big/(n + m)!\right]^{\frac{1}{2}}(\bar{C}_{n,m} - j\bar{S}_{n,m}) \quad .$$

Terms are included up to the ninth zonal harmonic and the 4,4 tesseral harmonic.

The accelerations due to the gravitational attractions of the sun and moon are given by

$$\mu_b\left[(\underline{r_b} - \underline{r})\big/|\underline{r_b} - \underline{r}|^3 - \underline{r_b}\big/|\underline{r_b}|^3\right]$$

b being 's' for the sun and 'm' for the moon.

Atmospheric drag is included if $ARMA \neq 0$. A modified version of the Jacchia 1965 model[5] is used to find the ambient air density. Firstly the subroutine computes the satellite's height and latitude (ϕ), the declination of the sun (δ) and the hour angle (H) of the satellite relative to the sun. The exospheric temperature (T_∞) is determined using the equation,

$$T_\infty = T_{NITE}\left[1 + 0.28\theta + 0.28\left[\cos\left\{\frac{\phi - \delta}{2}\right\}^{2.5} - \theta\right]\left[\cos\frac{\tau}{2}\right]^{2.5}\right] ,$$

where $\theta = \sin\dfrac{[\phi + \delta]^{2.5}}{2}$

and $\tau = \left[H - 0.78539816 + 0.20943951 \sin\left[1 + 0.78539816\right]\right]$.

The atmospheric temperature gradient at the reference altitude (120km) is given by:

$$T_{grad_0} = (T_\infty - 355)(0.029 \exp[- x^2/2])$$

where $x = \dfrac{T_\infty - 800}{750 + 1.722 \times 10^{-4}(T_\infty - 800)^2}$.

Subroutine MODAT is called to determine the atmospheric density ρ . The force acting on the satellite is given by $\frac{1}{2}\rho|\underline{V}|^2 SC_D$ where C_D is the drag coefficient, S is the effective cross-sectional area perpendicular to the air flow and $\underline{V}$ is the velocity of the satellite relative to the ambient air. $\underline{V}$ is given by

$$\underline{V} = \underline{v} - \underline{\omega}_a \times \underline{r}$$

where $\underline{r}$ and $\underline{v}$ are the position and velocity vectors of the satellite.

If the cartesian components of $\underline{V}$ are V_x, V_y, V_z then the contributions to the acceleration components are:

$$\ddot{x} = - \rho \frac{SC_D}{2M} |V| \underline{V_x} , \qquad \text{etc.}$$

where M is the mass of the satellite.

The variable ARMA is the quantity $\dfrac{SC_D}{2M} \times 1000$, the 1000 being a unit conversion factor.

The contribution to the acceleration due to the precession term is

$$p(\dot{x}\hat{z} - \dot{z}\underline{\hat{x}}) .$$

If required, the subroutine also evaluates the partial derivatives of the accelerations with respect to the initial position ($\underline{r}_0$) and velocity ($\underline{v}_0$) at epoch, i.e.

$$\frac{d^2}{dt^2}\left[\frac{\partial \underline{r}}{\partial(\underline{r}_0, \underline{v}_0)}\right] .$$

SUBROUTINE POSVEL

Summary	- The subroutine computes the geocentric cartesian components of position and velocity (POSVEL) of an earth satellite given the date, time and a set of 'PROP-type' orbital elements.
Language	- 1900 Fortran.
Author	- M.D. Palmer (February 1975).
Subroutine statement	- SUBROUTINE POSVEL (ELEMT, X, Y, Z, XDOT, YDOT, ZDOT, MJD, TIME, MJDOCH).
Input arguments	-
ELEMT(6,6)	Array of coefficients used to define mean orbital elements (see explanation).
MJD	Modified Julian day number of the given date.
TIME	Time, as a fraction of a day after MJD, at which the POSVEL is required.
MJDOCH	Modified Julian day number for 'PROP-type' orbital elements. Note: Normally MJD = MJDOCH and TIME = 0.0 .
Output arguments	-
X, Y, Z XDOT, YDOT, ZDOT	POSVEL.
Use of COMMON	- Certain arguments in the common block /CONST/ are used as follows:
Input arguments in /CONST/ -	
EMU	Earth's gravitational constant.
EJ2	Earth's second zonal harmonic, J_2 .
ERAD	Mean equatorial radius of the earth, R .
Source deck	- 115 cards including 18 comment cards (ICL code).
Local storage used	- 4 integer variables, 37 real variables and 60 real array elements.

Subordinate subprograms - Subroutine SHPAUX.

Explanation - The subroutine is essentially the same as subroutine SATXYZ used in the program PROP [6]. It will normally be used to produce a POSVEL from 'PROP-type' elements at the epoch corresponding to the elements, i.e. MJD = MJDOCH and TIME = 0.0 . The standard elements, eccentricity e , inclination, i , right ascension, Ω , argument of perigee, ω , and mean anomaly, M , are assumed to be polynomials in time of the form

$$e = e_0 + e_1 t + e_2 t^2 + e_3 t^3 + e_4 t^4 + e_5 t^5 .$$

The coefficients are stored in the array ELEMT(6,6). The first five rows contain six coefficients for e, i, Ω, ω and M and the first five elements of the sixth row are set to the values of the corresponding coefficients in the fourth-degree polynomial for mean motion.

The base elements (osculating elements with short and long periodic perturbations removed) are computed from the appropriate polynomials, whose degrees are specified by the elements of the array NOMIAL(6), set in a DATA statement. For the standard case mentioned above, i.e. $t = 0.0$, the elements of NOMIAL are set to zero. Long-periodic perturbations for this case are zero and so only the short-periodic terms need be incorporated in the computation which is identical to that in PROP.

Firstly $\bar{a}$ (semi-major axis), $\bar{q}$ $(= (1 - e^2)^{\frac{1}{2}})$, $\bar{E}$ (eccentric anomaly), $\bar{v}$ (true anomaly), $\bar{p}$ $(\bar{a}\bar{q}^2)$ and $\bar{u}$ $(\bar{v} + \bar{\omega})$ are computed. Subroutine SHPAUX then computes the short periodic perturbations. The formulae of section B.8 of Ref.17 are evaluated to obtain the POSVEL.

SUBROUTINE PRINT

Summary - The subroutine reads the unformatted disc output produced by subroutines OUTPUT and ORBINT and writes it to the line printer.

Language - 1900 Fortran.

Author - M.D. Palmer (January 1975).

Subroutine statement - SUBROUTINE PRINT (IS, NEL, NAP, NAE, K, STANAM).

Input arguments -

IS	The number of ground stations for which tables of look angles have been produced.
NEL	If NEL > 0, NEL = N + 1 where N is the number of lines of osculating elements stored on disc.
NAP	If NAP > 0, NAP = N + 1 where N is the number of lines in the table of apses stored on disc.
NAE	If NAE > 0, NAE = N + 1 where N is the number of lines in each table of look angles stored on disc.
K	The number of different sets of output held on disc channel 3, $0 \leqslant K \leqslant 5$.
STANAM(4,5)	Array, holding as text, the names of the ground stations for which look angles have been determined.

Use of COMMON - None.

Local storage used - 10 real variables and 4 integer variables.

Subordinate subprograms - None.

Source deck - 50 cards (ICL code).

Explanation - The subroutines OUTPUT and ORBINT store unformatted information on disc channels 3 and 4 respectively. PRINT recovers this information and writes it to the line printer on channel 2.

The information produced by OUTPUT falls into two categories, namely (a) tables of time, osculating elements and radial distance, and (b) tables of time, azimuth, elevation, range and range rate for up to four ground stations. The data is stored on disc channel 3 consecutively and in chronological order.

PRINT recovers and lists the information from each table separately, one line at a time. The table of osculating elements is recovered first followed by the tables of look angles. For the latter, only those lines in which the elevation >0.0 are actually printed.

ORBINT produces a table of apses which is written one line at a time to disc channel 4. PRINT recovers this information and writes it directly to the line printer. If no apses have been detected, an appropriate comment is printed.

A title is printed before each table.

SUBROUTINE RAZEL

Summary - The subroutine calculates the azimuth, elevation, range and range rate of an earth satellite for a given ground station.

Language - 1900 Fortran.

Author - M.D. Palmer (March 1975).

Subroutine statement - SUBROUTINE RAZEL (SD, I, XX, YY, ZZ, XD, YD, ZD, AZ, EL, R, RR).

Input arguments -

SD Sidereal angle at epoch (rad).

I Station number in the range $1 \leq I \leq 4$.

XX, YY, ZZ Satellite's geocentric position components (x,y,z) (km).

XD, YD, ZD Satellite's geocentric velocity components $(\dot{x},\dot{y},\dot{z})$ (km/s).

Output arguments -

AZ Azimuth (rad).

EL Elevation (rad).

R Range (km).

RR Range rate (km/s).

Use of COMMON - Certain arguments in the common blocks /STATION/ and /CONST/ are used as follows:

Input arguments in /STATION/ -

SL(4) Sin ϕ_i , where ϕ_i is the geodetic station latitude $(i = 1,4)$.

CL(4) Cos ϕ_i .

A1(4) $\dfrac{R_e^2 \cos \phi_i}{\left(R_e^2 \cos^2\phi_i + R_p^2 \sin^2\phi_i\right)^{\frac{1}{2}}} + h_i \cos \phi_i$ where R_p is the earth's polar radius, R_e is the earth's mean equatorial radius and h_i is the station's height above the geoid $(i = 1,4)$.

A2(4) $$\frac{R_p^2 \sin \phi_i}{\left(R_e^2 \cos^2 \phi_i + R_p^2 \sin^2 \phi_i\right)^{\frac{1}{2}}} + h_i \sin \phi_i \quad (i = 1,4).$$

SLONG(4) The geocentric station longitude, λ_i, rad (i = 1,4).

<u>Input argument in /CONST/</u> -

EOMEGA Mean rotation rate of the earth, ω_e (rad/s).

<u>Subordinate subprograms</u> - None.

<u>Source deck</u> - 19 cards (ICL code).

<u>Local storage used</u> - 5 real variables.

<u>Explanation</u> - POINT will produce, as an output option, an ephemeris of look angles (azimuth, elevation, range and range rate) for each of four ground stations. Each call to RAZEL gives a set of look angles for one station for a given point in time, the station being identified by the value of I ($1 \leqslant I \leqslant 4$).

Consider a cross-section of the earth containing the north and south poles and the station. Assume an axis system in this plane so that 0_x lies along the semi-major axis of the ellipse, 0_z along the minor axis and 0_y completes the right-handed set. The coordinates of a ground station are given by

$$x_s = \frac{R_e^2 \cos \phi}{\left(R_e^2 \cos^2 \phi + R_p^2 \sin^2 \phi\right)^{\frac{1}{2}}} + h \cos \phi$$

$$y_s = 0.0 \; .$$

$$z_s = \frac{R_p^2 \cos \phi}{\left(R_e^2 \cos^2 \phi + R_p^2 \sin^2 \phi\right)^{\frac{1}{2}}} + h \sin \phi$$

where ϕ is the station's geodetic latitude and h is its height above the spheroid. The satellite's coordinates (x,y,z) in PROP axes may be transformed into the above axis sytem by a rotation about the z-axis through an angle of $(\lambda + \theta)$, where λ is the station's geocentric longitude and θ is the sidereal angle. Thus the satellite's coordinates become

$$x' = x\cos(\lambda + \theta) + y\sin(\lambda + \theta)$$

$$y' = y\cos(\lambda + \theta) - x\sin(\lambda + \theta)$$

$$z' = z$$

and the range (r) of the satellite is given by

$$r = \left[(x' - x_s)^2 + (y' - y_s)^2 + (z' - z_s)^2\right]^{\frac{1}{2}} \quad .$$

The satellite's transformed velocity components are

$$\dot{x}' = \dot{x}\cos(\lambda + \theta) + \dot{y}\sin(\lambda + \theta) + \omega_e y$$

$$\dot{y}' = \dot{y}\cos(\lambda + \theta) - \dot{x}\sin(\lambda + \theta) - \omega_e z$$

$$\dot{z}' = \dot{z}$$

where ω_e is the earth's rotation rate. The satellite's range rate ($\dot{r}$) is given by

$$\dot{r} = (\dot{x}'x' + \dot{y}'y' + \dot{z}'z')/r \quad .$$

The azimuth (A) and elevation (E) of the satellite from the station are found in a 'local' axis sytem with the origin at the station, the x-axis along the local vertical, the z-axis along the local north and the y-axis in the direction of local east. The coordinates (x_ℓ, y_ℓ, z_ℓ) of the satellite in this system are:

$$x_\ell = (x' - x_s)\cos\phi + (z' - z_s)\sin\phi$$

$$y_\ell = y'$$

$$z_\ell = (z' - z_s)\cos\phi - (x' - x_s)\sin\phi \quad .$$

Thus

$$A = \mathrm{ATAN2}(y_\ell, z_\ell)$$

and

$$E = \mathrm{ATAN2}\left[x_\ell, (y_\ell^2 + z_\ell^2)^{\frac{1}{2}}\right] \quad .$$

A rigorous derivation of this theory is given in Ref.18.

SUBROUTINE READEL

Summary - The subroutine reads RAE orbital elements from a set of five punched cards and stores them in a standard array.

Language - 1900 Fortran.

Author - M.D. Palmer (March 1975).

Subroutine statement - SUBROUTINE READEL (ELEMT, MJDOCH).

Output arguments -

ELEMT(6,6) Array of coefficients used to define orbital elements (see explanation).

MJDOCH MJD of the midnight epoch at which the elements are defined.

Use of COMMON - None.

Source deck - 15 cards (ICL code).

Local storage used - 6 real array elements, 3 real variables and 3 integer variables.

Subordinate subprograms - None.

Explanation - The subroutine is essentially the same as the subroutine ELREAD used in the program PROP [6]. It is assumed that the five basic orbital elements of an earth satellite, e, i, Ω, ω, M may be represented by polynomials in time of degree five, e.g.

$$e = e_0 + e_1 t + e_2 t^2 + e_3 t^3 + e_4 t^4 + e_5 t^5 \quad .$$

The six coefficients for each polynomial are read from one card and stored in one row of the array ELEMT. The first five elements of the sixth row are set to the values of the corresponding coefficients in the fourth degree polynomial in mean motion as follows:

$$\text{ELEMT}(6,j) = j\ \text{ELEMT}(5,j+1) \qquad \text{for } j = 1,5 \quad .$$

The coefficients, as read, have units of degrees and days and these are changed to radians and seconds before storage.

The five data cards must contain MJDOCH in cols. 1 to 5 and a serial number (1,...,5) in col.7. A check is made that the cards are in the correct sequence and, if not, a STOP99 instruction is obeyed.

SUBROUTINE REIP

Summary - The subroutine reads the input data for POINT and, where necessary, converts it to the form required by the main program.

Language - 1900 Fortran.

Author - M.D. Palmer (January 1975).

Subroutine statement - SUBROUTINE REIP.

Use of COMMON - Certain quantities in the common blocks /CINTEG/, /CONST/, /INTP/,/PETURB/ and /STATION/ are used as follows:

Input arguments in /CONST/ -

EMU	Earth's gravitational constant, μ (km^3/s).
ERAD	Mean equatorial radius of the earth, R (km).
EOMEGA	Mean rotation rate of the earth ω_e in rad/s.

Input arguments in /PETURB/ -

HALFCD	The quantity $\frac{1}{2}C_D$ where C_D is the satellite's drag coefficient.

Output arguments in /CINTEG/ -

MJD	Modified Julian day number of epoch.
NM	Number of equations to be integrated.
X, Y, Z	Cartesian components of the satellite's geocentric position (x,y,z) at epoch (km).
XD, YD, ZD	Cartesian components of the satellite's geocentric velocity $(\dot{x},\dot{y},\dot{z})$ at epoch (km/s).

Output arguments in /INTP/ -

EP	Time of epoch, in fractions of a day relative to MJDOCH.
DT	Interpolation inteval, min.
TINT	The time from epoch at which the first interpolation is required, hours.
TIOR	Integration period, hours.

NPV	NPV = 1 if an ephemeris of geocentric position and velocity components (POSVEL) is required.
NEL	NEL = 1 if an ephemeris of osculating elements is required.
NAP	NAP = 1 if a table of apses is required.
NAE	NAE = 1 if tables of look angles are required.
COV(6,6)	Covariance matrix corresponding to the POSVEL.
NOTPT	NOTPT = 1 if an ephemeris of POSVEL and, if specified, the propagated covariance matrix is to be written to a permanent disc file.

Output arguments in /PETURB/ -

ARMA	The product AM × HALFCD × 1000.0, where AM is the area-to-mass ratio of the satellite in m^2/kg. If ARMA $\neq$ 0.0, drag terms are included in the integration.
TNITE	Nightime minimum exospheric temperatures (K).
SMEAN	Mean value over three days of the solar 10.7cm radiation flux in units of $10^{-22} W\ m^{-2}\ Hz^{-1}$.
SOLBAR	Mean value over three solar rotations of the 10.7cm radiation flux in units of $10^{-22} W\ m^{-2}\ Hz^{-1}$.
RATE	Angular velocity of the atmosphere, rad/s.
LSP	LSP = 1 if luni-solar perturbations are included in the integration.
NSR	A spare variable, to be used as a flag should solar radiation pressure terms be incorporated in the integration.

Output arguments in /STATION/ -

K	The number of ground stations for which tables of look angles are required (K $\leq$ 4).
SLT(4)	Sin ϕ_i , where ϕ_i is the geodetic station latitude (i = 1,...,4).
CLT(4)	Cos ϕ_i (i = 1,...,4).

A1(4) $$\frac{R_e^2 \cos \phi_i}{\left(R_e^2 \cos^2\phi_i + R_p^2 \sin^2\phi_i\right)^{\frac{1}{2}}} + h_i \cos \phi_i$$ where R_p and R_e are the earth's polar and mean equatorial radii and h_i is the station's height above the geoid $(i = 1,\ldots,4)$.

A2(4) $$\frac{R_p^2 \sin \phi_i}{\left(R_e^2 \cos^2\phi_i + R_p^2 \sin^2\phi_i\right)^{\frac{1}{2}}} + h_i \sin \phi_i \ .$$

SLO(4) The geocentric station longitude λ_i , rad $(i = 1,\ldots,4)$.

STANAM(4,5) Array holding the ground station names as text.

Input and output arguments in /CINTEG/ -

H On input, the integration step length as set in BLOCK DATA. On output H/C1 (C1 $\neq$ 0) where C1 is an input data item.

Subordinate subprograms - The subroutines ELTOCO, IPTOCO, MATMUL, POSVEL, RLEASE, ROTATE, SDC2EL, SETPD, SHPAUX and TRAMAT and the functions ANGLE and EAFKEP.

Source deck - 130 cards (ICL code).

Local storage used - 113 real array elements, 18 real variables and 2 integer variables.

Explanation - The subroutine determines the position and velocity of an earth satellite at a given epoch in terms of its geocentric cartesian components in the PROP axes system. The components may be read directly or input in an alternative form and converted. The four alternative forms are:

(i) a standard set of launch vehicle injection conditions (speed, climb angle, azimuth, radius, latitude and longitude);

(ii) a set of osculating elements (semi-major axis, eccentricity, inclination, right ascension of the ascending node, argument of perigee and mean anomaly);

(iii) a set of USAF Space Defense Center (SDC) two line-elements; and

(iv) a set of RAE five-line mean elements.

A choice of the perturbations to be included in the integration is provided and there are a number of output formats.

One option is to propagate the covariance matrix associated with the position and velocity components. The initial covariance matrix may be provided either

(i) in terms of the position and velocity components (km and km/s)

or (ii) in terms of injection conditions with units of ft/s, degrees and feet. In this case, the orbit must also be supplied as a set of injection conditions.

In the latter case, the subroutine carries out a transformation of the form:

$$\mathrm{cov}(\underline{X_0}) = \underline{A}\ \mathrm{cov}(\underline{Y_0})\ \underline{AT}$$

where $\mathrm{cov}(\underline{X_0})$ is the covariance matrix in terms of position and velocity components,

$\mathrm{cov}(Y_0)$ is the injection covariance matrix

and $\underline{A}$ is the matrix of partial derivatives given by

$$\begin{bmatrix}
\frac{\partial x}{\partial v} & \frac{\partial x}{\partial \theta} & \frac{\partial x}{\partial \psi} & \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \phi} & \frac{\partial x}{\partial \lambda'} \\
\frac{\partial y}{\partial v} & \frac{\partial y}{\partial \theta} & \frac{\partial y}{\partial \psi} & \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \phi} & \frac{\partial y}{\partial \lambda'} \\
\frac{\partial z}{\partial v} & \frac{\partial z}{\partial \theta} & \frac{\partial z}{\partial \psi} & \frac{\partial z}{\partial r} & \frac{\partial z}{\partial \phi} & \frac{\partial z}{\partial \lambda'} \\
\frac{\partial \dot{x}}{\partial v} & \frac{\partial \dot{x}}{\partial \theta} & \frac{\partial \dot{x}}{\partial \psi} & \frac{\partial \dot{x}}{\partial r} & \frac{\partial \dot{x}}{\partial \phi} & \frac{\partial \dot{x}}{\partial \lambda'} \\
\frac{\partial \dot{y}}{\partial v} & \frac{\partial \dot{y}}{\partial \theta} & \frac{\partial \dot{y}}{\partial \psi} & \frac{\partial \dot{y}}{\partial r} & \frac{\partial \dot{y}}{\partial \phi} & \frac{\partial \dot{y}}{\partial \lambda'} \\
\frac{\partial \dot{z}}{\partial v} & \frac{\partial \dot{z}}{\partial \theta} & \frac{\partial \dot{z}}{\partial \psi} & \frac{\partial \dot{z}}{\partial r} & \frac{\partial \dot{z}}{\partial \phi} & \frac{\partial \dot{z}}{\partial \lambda'}
\end{bmatrix}$$

where x, y, z are the position components,

$\dot{x}, \dot{y}, \dot{z}$ the velocity components,

v the speed at injection,

θ the climb angle,

ψ the azimuth,

	r	the radial distance,
	ϕ	the geocentric latitude
and	λ'	the longitude in inertial space.

λ' is defined by $\lambda' = \hat{\theta} + \lambda$ where λ is the geocentric longitude and $\hat{\theta}$ is the modified sidereal angle given by

$$\hat{\theta} = 100.075542 + 360.985612288(t - 33282.0)$$

where $t = \text{MJD} + \text{EP}$.

It can be shown that:

Row 1

$$\frac{\partial x}{\partial v} = \frac{\partial x}{\partial \theta} = \frac{\partial x}{\partial \psi} = 0.0\ , \qquad \frac{\partial x}{\partial r} = \cos\phi\cos\lambda'$$

$$\frac{\partial x}{\partial \phi} = -\,r\sin\phi\cos\lambda'\ , \qquad \frac{\partial x}{\partial \lambda'} = -\,r\cos\phi\sin\lambda'$$

Row 2

$$\frac{\partial y}{\partial v} = \frac{\partial y}{\partial \theta} = \frac{\partial y}{\partial \psi} = 0.0\ , \qquad \frac{\partial y}{\partial r} = \cos\psi\sin\lambda'$$

$$\frac{\partial y}{\partial \phi} = -\,r\sin\phi\sin\lambda'\ , \qquad \frac{\partial y}{\partial \lambda'} = r\cos\phi\cos\lambda'$$

Row 3

$$\frac{\partial z}{\partial v} = \frac{\partial z}{\partial \theta} = \frac{\partial z}{\partial \psi} = 0.0\ , \qquad \frac{\partial z}{\partial r} = \sin\phi$$

$$\frac{\partial z}{\partial \phi} = r\cos\phi\ , \qquad \frac{\partial z}{\partial \lambda'} = 0.0$$

Row 4

$$\frac{\partial \dot{x}}{\partial v} = \sin\theta\cos\phi\cos\lambda' - \cos\theta\,[\cos\psi\sin\phi\cos\lambda' + \sin\psi\sin\lambda']$$

$$\frac{\partial \dot{x}}{\partial \theta} = v[\cos\theta\cos\phi\cos\lambda' + \sin\theta\,[\cos\psi\sin\phi\cos\lambda' + \sin\psi\sin\lambda']]$$

$$\frac{\partial \dot{x}}{\partial \psi} = v\cos\theta\,[\sin\psi\sin\phi\cos\lambda' - \cos\psi\sin\lambda']$$

$$\frac{\partial \dot{x}}{\partial r} = 0.0$$

$$\frac{\partial \dot{x}}{\partial \phi} = v[- \sin\theta \sin\phi \cos\lambda' - \cos\theta \cos\psi \cos\phi \cos\lambda']$$

$$\frac{\partial \dot{x}}{\partial \lambda'} = v[- \sin\theta \cos\phi \sin\lambda' + \cos\theta [\cos\psi \sin\phi \sin\lambda' - \sin\psi \cos\lambda']]$$

<u>Row 5</u>

$$\frac{\partial \dot{y}}{\partial v} = \sin\theta \cos\phi \sin\lambda' + \cos\theta [\sin\phi \cos\lambda' - \cos\psi \sin\phi \sin\lambda']$$

$$\frac{\partial \dot{y}}{\partial \theta} = v [\cos\theta \cos\phi \sin\lambda' - \sin\theta [\sin\psi \cos\lambda' - \cos\psi \sin\phi \sin\lambda']]$$

$$\frac{\partial \dot{y}}{\partial \psi} = v \cos\theta [\cos\psi \cos\lambda' + \sin\phi \sin\psi \sin\lambda']$$

$$\frac{\partial \dot{y}}{\partial r} = 0.0$$

$$\frac{\partial \dot{y}}{\partial \phi} = v[- \sin\theta \sin\phi \sin\lambda' - \cos\theta \cos\psi \cos\phi \sin\lambda']$$

$$\frac{\partial \dot{y}}{\partial \lambda'} = v [\sin\theta \cos\phi \cos\lambda' + \cos\theta [- \sin\psi \sin\lambda' - \cos\psi \sin\phi \cos\lambda']]$$

<u>Row 6</u>

$$\frac{\partial \dot{z}}{\partial v} = \sin\theta \sin\phi + \cos\theta \cos\psi \cos\phi$$

$$\frac{\partial \dot{z}}{\partial \theta} = v \cos\theta \sin\phi - v \sin\theta \cos\psi \cos\phi$$

$$\frac{\partial \dot{z}}{\partial \psi} = - v \cos\theta \sin\psi \cos\phi$$

$$\frac{\partial \dot{z}}{\partial r} = 0.0$$

$$\frac{\partial \dot{z}}{\partial \phi} = v \sin\theta \cos\phi - v \cos\theta \cos\psi \sin\phi$$

$$\frac{\partial \dot{z}}{\partial \lambda'} = 0.0 \; .$$

SUBROUTINE ROTATE

Summary - The subroutine performs rotations, about the z-axis by a given angle at a given angular velocity.

Language - ASA Fortran (Standard Fortran 4).

Author - A.W. Odell (August 1970).

Subroutine statement - SUBROUTINE ROTATE (X, Y, XDOT, YDOT, ANGLE, ANGVEL).

Input arguments -

ANGLE	Angle θ , through which coordinates are to be rotated - anticlockwise.
ANGVEL	Angular velocity, ω , to be applied to velocity coordinates - anticlockwise.

Input and output arguments -

X	x coordinate.
Y	y coordinate.
XDOT	$\dot{x}$ coordinate.
YDOT	$\dot{y}$ coordinate.

Use of COMMON - None.

Source deck - 11 cards (ICL code).

Local storage used - 3 real variables.

Subordinate subprograms - None.

Explanation - The rotation is equivalent to the following

$$(x,y) \rightarrow (x \cos\theta - y \sin\theta \; , \; y \cos\theta + x \sin\theta)$$

followed by

$$(\dot{x},\dot{y}) \rightarrow (\dot{x} \cos\theta - \dot{y} \sin\theta - \omega y \; , \; \dot{y} \cos\theta + \dot{x} \sin\theta + \omega x) \; .$$

FUNCTION SCPROD

Summary - The function gives the scalar (inner) product of two arrays.

Language - PLAN, for use with 1900 Fortran.

Author - A.W. Odell (March 1973).

Function statement - FUNCTION SCPROD (A, B, NA, NB, N).

Input arguments -

A(1), B(1) Locations of the first elements to be multiplied (must be array elements).

NA, NB Increments in the arrays, A, B of elements to be multiplied.

N Number of elements to be multiplied.

Output function -

SCPROD The scalar product $\sum_{i=1}^{N} A(1 + (i - 1)NA)B(1 + (i - 1)NB)$.

Use of COMMON - None.

Source deck - 25 cards, including 1 comment card (ICL code).

Local storage used - 2 words for variables, plus 22 words for program.

Subordinate subprograms - None.

Explanation - The arrays in the calling routine may have any dimensions although they are treated as one-dimensional arrays in the function. For simplicity A and B are dimensioned '1'. This may cause the function to fail in TRACE2, so it should not be compiled in this mode. If $N \leqslant 0$, the function will return zero.

SUBROUTINE SDC2EL

Summary - The subroutine reads a set of Space Defense Center (SDC) two-line elements and converts them to an equivalent set of 'PROP' type elements.

Language - 1900 Fortran.

Author - M.D. Palmer (December 1974).

Subroutine statement - SUBROUTINE SDC2EL (MJDOCH, ELEMT).

Output arguments -

MJDOCH	Modified Julian day number of epoch.
ELEMT(6,6)	Array of PROP-type orbital elements.

Use of COMMON - Certain arguments in the common block /CONST/ are used as follows:

Input arguments in /CONST/ -

EMU	Earth's gravitational constant, μ (km^3/s^2).
XJ2	Earth's second zonal harmonic coefficient, J_2 .
XJ3	Earth's third zonal harmonic coefficient, J_3 .
ERAD	Earth's mean equatorial radius, R (km).

Subordinate subprograms - None.

Local storage used - 5 integer variables and 16 real variables.

Source deck - 71 cards including 10 comment cards (ICL code).

Explanation - This subroutine is essentially the subroutine SDCELS used in the program PROP [6]. A set of orbital elements consists of eccentricity, e , inclination, i , right ascension of ascending node, Ω , argument of perigee, ω , mean anomaly, M , and mean motion, n . With the exception of mean motion, these elements may be assumed to be polynomials in time, i.e.

$$e = e_0 + e_1 t + e_2 t^2 + \dots \text{ etc.},$$

where e_1, e_2, etc. are the 'rate elements'.

SDC elements differ from PROP elements in the following ways:

(i) PROP elements are for midnight epochs, whereas SDC elements relate to an ascending node.

(ii) PROP elements follow the long-periodic motion of the satellite, but SDC elements have been 'corrected' for long-periodic perturbations associated with a nominal value of the earth's third zonal harmonic.

(iii) The origin of the right ascension of the ascending node is the mean equinox of date for SDC but the 1950.0 equinox for PROP.

(iv) SDC elements i_0, Ω_0, ω_0 are in degrees and the SDC units for the M-polynomial quantities are revolutions and days, whereas PROP elements are stored in units of radians and seconds.

The quantities read from the first SDC card are card number, satellite identification number, year of epoch, day of year, fraction of day, M_2 and M_3. It is read in the format

(I1, 1X, I5, 2X, A8, 1X, I2, I3, F9.8, 1X, F10.8, 1X, E8.5) .

The second card contains card number, satellite, i_0, Ω_0, e_0, ω_0, M_0 and M_1 and is read in the format

(I1, 1X, I5, 2(1X, F8.4), 1X, F7.7, 2(1X, F8.4), 1X, F11.8) .

The semi-major axis, which is an auxiliary element, is computed from

$$a = a_{osc}\left[1 - \frac{J_2}{4}\left(\frac{R}{a_{osc}}\right)^2\left(2 - 3\sin^2 i_0\right)\left(1 - e_0^2\right)^{-3/2}\right]$$

where the osculating semi-major axis, a_{osc}, is given by $a_{osc} = \left[\mu/n_0^2\right]^{1/3}$ where n_0 is M_1 converted to rad/s.

Ω_1 and ω_1 are computed from

$$\Omega_1 = -\frac{P}{2}\cos i_0 \qquad \text{and} \qquad \omega_1 = P\left[1 - \frac{5}{4}\sin^2 i_0\right]$$

where $P = 3J_2\left[\frac{R}{a(1 - e_0^2)}\right]^2 n_0$.

The time in days, τ , by which the new epoch is later than the epoch of the SDC elements is found. Ω_0, ω_0 and M_0 are replaced by $\Omega_0 + \Omega_1\tau$, $\omega_0 + \omega_1\tau$ and $M_0 + M_1\tau + M_2\tau^2 + M_3\tau^3$, respectively, and M_1 and M_2 by $M_1 + 2M_2\tau + 3M_3\tau^2$ and $M_2 + 3M_3\tau$. Coefficients of the mean-motion polynomial are obtained from $n_j = (j + 1)M_{j+1}$ for $j = 0, 1, 2$ and 3.

Ω_0 is decreased by $3.506 \times 10^{-5}D$ where D is the number of days elapsed from 1950.0, to allow for equinox conversion.

The SDC long-periodic perturbations are computed from

$$L_L = -\frac{K_3}{4}\left[\frac{3 + 5c}{1 + c}\right]e_0 \cos \omega_0$$

and

$$a_{yNL} = -K_3/2 ,$$

where $c = \cos i_0$
and $K_3 = J_3 \sin i_0/J_2a(1 - e_0^2)$.

M_0 is replaced by $M_0 + L_L$, e_0 by $[C^2 + S^2]^{\frac{1}{2}}$ and ω_0 by $\tan^{-1}(S/C)$, where $C = e_0 \cos \omega_0$ and $S = e_0 \sin \omega_0 + a_{yNL}$. M_0 is adjusted such that $M_0 + \omega_0$ remains constant.

The derived PROP elements are stored in the following positions of the array ELEMT: e_0, i_0, ω_0, M_0 and n_0 make up the first column; M_0 , in the fifth row, is followed by M_1, M_2 and M_3 ; n_0 in the sixth row is followed by n_1 and n_2 . All other elements of ELEMT are set to zero.

SUBROUTINE SETPD

Summary - The subroutine sets the initial, non-zero values of the partial derivatives $\partial\underline{r}/\partial(\underline{r}_0,\underline{v}_0)$ and $\frac{d}{ds}\left[\frac{\partial\underline{r}}{\partial(\underline{r}_0,\underline{v}_0)}\right]$.

Language - 1900 Fortran.

Author - M.D. Palmer (February 1973).

Subroutine statement - SUBROUTINE SETPD.

Use of COMMON - Certain arguments in the common block /CINTEG/ are used as follows:

Input argument -

TVEL The initial value of dt/ds.

Output arguments -

PD(18) Array holding the partial derivatives of position $\partial\underline{r}/\partial(\underline{r}_0,\underline{v}_0)$.

PDVEL(18) Array holding partial derivatives $\frac{d}{ds}\left[\frac{\partial\underline{r}}{\partial(\underline{r}_0,\underline{v}_0)}\right]$.

Source deck - 8 cards (ICL code).

Local storage used - 1 integer variable.

Subordinate subprograms - None.

Explanation - Before using DEQRSPT to integrate the 22 equations of motion, the initial non-zero values of two partial derivative matrices must be set as follows:

$$\frac{\partial\underline{r}}{\partial(\underline{r}_0,\underline{v}_0)} = \begin{bmatrix} PD(1) & \ldots & PD(6) \\ PD(7) & \ldots & PD(12) \\ PD(13) & .. & PD(18) \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial x}{\partial x_0} & \frac{\partial x}{\partial y_0} & \frac{\partial x}{\partial z_0} & \frac{\partial x}{\partial \dot{x}_0} & \frac{\partial x}{\partial \dot{y}_0} & \frac{\partial x}{\partial \dot{z}_0} \\ \frac{\partial y}{\partial x_0} & \frac{\partial y}{\partial y_0} & \frac{\partial y}{\partial z_0} & \frac{\partial y}{\partial \dot{x}_0} & \frac{\partial y}{\partial \dot{y}_0} & \frac{\partial y}{\partial \dot{z}_0} \\ \frac{\partial z}{\partial x_0} & \frac{\partial z}{\partial y_0} & \frac{\partial z}{\partial z_0} & \frac{\partial z}{\partial \dot{x}_0} & \frac{\partial z}{\partial \dot{y}_0} & \frac{\partial z}{\partial \dot{z}_0} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

and

$$\frac{d}{ds}\left[\frac{\partial \underline{r}}{\partial(\underline{r_0},\underline{v_0})}\right] = \begin{bmatrix} \text{PDVEL}(1) & \ldots & \text{PDVEL}(6) \\ \text{PDVEL}(7) & \ldots & \text{PDVEL}(12) \\ \text{PDVEL}(13) & \ldots & \text{PDVEL}(18) \end{bmatrix} \left[\frac{dt}{ds}\right]$$

$$= \begin{bmatrix} \frac{\partial \dot{x}}{\partial x_0} & \frac{\partial \dot{x}}{\partial y_0} & \frac{\partial \dot{x}}{\partial z_0} & \frac{\partial \dot{x}}{\partial \dot{x}_0} & \frac{\partial \dot{x}}{\partial \dot{y}_0} & \frac{\partial \dot{x}}{\partial \dot{z}_0} \\ \frac{\partial \dot{y}}{\partial x_0} & \frac{\partial \dot{y}}{\partial y_0} & \frac{\partial \dot{y}}{\partial z_0} & \frac{\partial \dot{y}}{\partial \dot{x}_0} & \frac{\partial \dot{y}}{\partial \dot{y}_0} & \frac{\partial \dot{y}}{\partial \dot{z}_0} \\ \frac{\partial \dot{z}}{\partial x_0} & \frac{\partial \dot{z}}{\partial y_0} & \frac{\partial \dot{z}}{\partial z_0} & \frac{\partial \dot{z}}{\partial \dot{x}_0} & \frac{\partial \dot{z}}{\partial \dot{y}_0} & \frac{\partial \dot{z}}{\partial \dot{z}_0} \end{bmatrix} \left[\frac{dt}{ds}\right]$$

$$= \begin{bmatrix} 0 & 0 & 0 & dt/ds & 0 & 0 \\ 0 & 0 & 0 & 0 & dt/ds & 0 \\ 0 & 0 & 0 & 0 & 0 & dt/ds \end{bmatrix}$$

where s is the independent variable given initially by $dt/ds = r_0^{3/2}\mu^{-1/2}$, $\underline{r_0} = (x_0, y_0, z_0)$ being the initial radius vector to the satellite and, $\underline{v_0} = (\dot{x}_0, \dot{y}_0, \dot{z}_0)$ the initial velocity of the satellite and μ the earth's gravitational constant.

SUBROUTINE SHPAUX

Summary - The subroutine calculates the short-periodic perturbations, due to the zonal harmonic J_2, in an earth satellite's orbit.

Language - 1900 Fortran.

Author - M.D. Palmer (January 1975).

Subroutine statement - SUBROUTINE SHPAUX (ECC, SI, CI, OMEGA, V, SV, CV, U, FMANOM, F, P, POVERR, Q, TSP, EJ2, ERAD, EMU).

Input arguments -

ECC	Eccentricity, e .
SI	Sin i , where i is the inclin tion.
CI	Cos i .
OMEGA	Argument of perigee, ω (rad).
V	True anomaly, v (rad).
SV	Sin v .
CV	Cos v .
U	Argument of latitude, u (rad).
FMANOM	Fractional part of mean anomaly, M_f (rad).
F	$\sin^2 i$.
P	Semi-latus rectum, p (km).
POVERR	p/r , where r is the satellite's radial distance (km).
Q	$(1 - e^2)^{\frac{1}{2}}$.
EJ2	Earth's second zonal harmonic, J_2 .
ERAD	Mean equatorial radius of the earth (km).
EMU	Earth's gravitational constant, μ (km^3/s^2).

Output arguments -

TSP(6)	Array of short-periodic perturbations, (see explanation).

Use of common - None.

Source deck - 30 cards including 3 comment cards (ICL code).

Local storage used - 14 real variables.

Subordinate subprograms - None.

Explanation - The subroutine is essentially the same as the subroutine SHOPER used in the program PROP [6]. It computes the short-periodic perturbations required by the subroutine POSVEL. Input elements are assumed to be the mean elements of Merson[19], and the expressions for the short-periodic perturbations are:

$$di_s = \tfrac{1}{2}KC \sin i \cos i \quad ,$$

$$d\Omega_s \sin i = -K(v - M_f + e \sin v - \tfrac{1}{2}S) \sin i \cos i \quad ,$$

$$du_s + d\Omega_s \cos i = Kh\left[v - M_f + e \sin v + \tfrac{1}{3}ge \sin v\,(g + \cos v)\right] + \tfrac{1}{3}Kf\left[\tfrac{1}{4} \sin 2u + e \sin (2u - v)\right] \quad ,$$

$$dp_s/2p = \tfrac{1}{2}K(hq + fc) \quad ,$$

$$dr_s/p = \tfrac{1}{3}K\left[\tfrac{1}{2} f \cos 2u - h(1 + g \cos v - qr/p)\right]$$

and

$$d\dot{r}_s = \tfrac{1}{3}(\mu/p)^{\frac{1}{2}}K\left[h \sin v\,(-\tfrac{1}{2}qe + g(p/r)^2) - f(p/r)^2 \sin 2u\right]$$

where $K = 1\tfrac{1}{2}J_2(R/p)^2$,

$C = \cos 2u + e\left[\cos (2u - v) + \tfrac{1}{3} \cos (2u + v)\right]$,

$S = \sin 2u + e\left[\sin (2u - v) + \tfrac{1}{3} \sin (2u + v)\right]$,

$h = 1 - 1\tfrac{1}{2}f$

and $g = e/(1 + q)$.

The perturbations are stored, in the above order, in the array TSP.

SUBROUTINE SMPOS

Summary - The subroutine computes the geocentric cartesian coordinates (km) of the sun and moon at a given time.

Language - ASA Fortran (Standard Fortran 4).

Author - A.W. Odell (May 1973).

Subroutine statement - SUBROUTINE SMPOS.

Dummy arguments - None.

Use of COMMON - Certain quantities in common block /CSMOON/ are used as follows:

Input arguments in /CSMOON/ -

MJDT	Modified Julian day number of the required time.
TIMET	Time, as a fraction of a day, since 0 hours on MJDT.

Output arguments in /CSMOON/ -

XS, YS, ZS	The cartesian coordinates $(x_s,y_s,z_s) = r_s$ of the sun.
XM, YM, ZM	The cartesian coordinates $(x_m,y_m,z_m) = r_m$ of the moon.

Input and output argument in /CSMOON/ -

TABLE(43)	Working space.

Source deck - 27 cards including 1 comment card (ICL code).

Local storage used - 2 integer variables, 1 logical variable.

Subordinate subprograms - The subroutine UTD4, and the functions INFIND and SCPROD.

Explanation - The subroutine obtains the sun-moon coordinates by interpolation in a table of daily positions, with 2nd and 4th differences, using Everett's interpolation formula. The table is read from a disc file when required, using the subroutine UTD4. It is stored in an area called SUNMOONTABLE, the location of which is found using the function INFIND. The area is contained in a disc file, which must have been opened, before entry to SMPOS, as peripheral unit ED7.

The table is stored as follows: firstly, the modified Julian day numbers of the first and last sets of coordinates (2 integers), then data for each midnight as follows (sets of 18 reals): r_s, r_m, $\Delta^2 r_s$, $\Delta^2 r_m$, $\Delta^4 r_s$, $\Delta^4 r_m$.

If f_0, f_1 denote values of f at $t = 0$ and $t = 1$, Everett's interpolation formula to the 4th differences[20] gives:

$$f(t) = D_0 + t\Big(D_1 + (t - 1)\Big(D_2 + (t + 1)\Big(D_3 + (t - 2)\Big(D_4 + (t + 2)D_5\Big)\Big)\Big)\Big) \quad ,$$

where $D_{2n} = \delta^{2n} f_0/(2n)!$, $D_{2n+1} = \delta^{2n}(f_1 - f_0)/(2n + 1)!$.

The data was originally obtained on punched cards from JPL [3,4] and has been transformed, before storing on the disc, into the SAO/PROP [6] system of axes, using the subroutine AX1950. If the time input to SMPOS lies outside the range of data stored on the disc, a STOP77 statement will be obeyed.

FUNCTION SOLVIN

Summary	- Given a function and its derivative at two points, the function solves four problems involving cubic interpolation.
Language	- ASA Fortran (Standard Fortran 4).
Author	- A.W. Odell (January 1975).
Function statement	- FUNCTION SOLVIN (MODE, M, FO, F1, FDO, FD1, F).
Input arguments	-
MODE	Number specifying problem to be solved, see explanation.
H	Difference, h, between the arguments of the function (i.e. $h = x_1 - x_0$).
FO, F1	Function values f_0 at $x = x_0$ and f_1 at $x = x_1$.
FDO, FD1	Derivative values f'_0, at $x = x_0$ and f'_1 at $x = x_1$.
F	Required function value or argument - see explanation.
Output function	-
SOLVIN	Solution to problem - see explanation.
Use of COMMON	- None.
Source deck	- 33 cards, including 5 comment cards (ICL code).
Local storage used	- 9 real variables.
Subordinate subprograms	- None.

Explanation - A cubic polynomial $P(x) = f_0 + f'_0 x + a_2 x^2 + a_3 x^3$ is first fitted to the data giving $a_2 = (3B - A)/h$ and $a_3 = (A - 2B)/h^2$ where $A = f' - f_0$ and $B = (f_1 - f_0)/h - f'_0$.

Then,

(1) if MODE = 1, Newton's method is used to find the value of x for which $P(x) = F$,

(2) if MODE = 2, Newton's method is used to find the value of x for which $P'(x) = F$,

(3) if MODE = 3, $P(F)$ is evaluated, and

(4) if MODE = 4, $P'(F)$ is evaluated.

SUBROUTINE TRAMAT

Summary - The subroutine performs matrix transposition.

Language - 1900 Fortran.

Author - M.D. Palmer (May 1973).

Subroutine statement - SUBROUTINE TRAMAT (A, AT, K, L).

Input arguments -

A	The matrix to be transposed.
K	Number of rows in A.
L	Number of columns in A.

Output arguments

AT	The transposed matrix.

Use of COMMON - None.

Source deck - 7 cards (ICL code).

Local storage used - 2 integer variables.

Subordinate subprograms - None.

Explanation - The subroutine interchanges the rows and columns of a matrix A to form the transpose A^T, i.e.

$$A^T(i,j) = A(j,i) .$$

SUBROUTINE TRINV

Summary - The subroutine obtains polar coordinates from (two-dimensional) cartesian coordinates.

Language - ASA Fortran (Standard Fortran 4).

Author - R.H. Gooding (May 1968).

Subroutine statement - SUBROUTINE TRINV (Y, X, R, TH).

Input arguments -

Y Cartesian y-coordinate (arbitrary).

X Cartesian x-coordinate (arbitrary).

Output arguments -

R Polar r-coordinate.

TH Polar θ-coordinate.

Use of COMMON - None.

Source deck - 8 cards, including 2 comment cards (ICL code).

Local storage used - None.

Subordinate subprograms - None.

Explanation - The equations $r \cos \theta = x$, and $r \sin \theta = y$ are solved for r and θ, using the standard ATAN2 function to give θ between $-\pi$ and $+\pi$. If $x = y = 0$ the ATAN2 function is not used and θ is set to zero.

Remarks -

(1) The subroutine, if used twice, provides a convenient solution of the three-dimensional cartesian-to-polar transformation. Thus to solve the equations,

$$r \sin \theta \cos \phi = x ,$$

$$r \sin \theta \sin \phi = y ,$$

and

$$r \cos \theta = z$$

for r, θ and ϕ, the following two statements will suffice:

CALL TRINV (Y, X, RSINTH, PHI) , giving $r \sin \theta$ and ϕ

and

CALL TRINV (RSINTH, Z, R, TH) , giving r and θ .

Moreover, this solution will give maximum accuracy for θ and ϕ, with both angles set in the correct quadrant.

(2) The actual input and output arguments must be distinct.

REFERENCES

No.	Author	Title, etc.
1	G.E. Cook M.D. Palmer	Orbital requirements for an Astronomical Roentgen Observatory Satellite. J. Brit. Interplan. Soc. 28, 763 (1975) RAE Technical Report 74110 (1974)
2	R.H. Merson	Numerical integration of the differential equations of celestial mechanics. RAE Technical Report 74184 (1974)
3	P.R. Peabody J.F. Scott E.G. Orozco	Users' description of the JPL ephemeris tapes. JPL Technical Report 32-580 (1964)
4	P.R. Peabody J.F. Scott E.G. Orozco	JPL ephemeris tapes E9510, E9511 and E9512. JPL Technical Memorandum 33-167 (1964)
5	G.E. Cook	The semi-annual variation in the upper atmosphere: a review. Annales de Géophysique 25, 451 (1969) RAE Technical Report 69074 (1969)
6	R.H. Gooding R.J. Taylor	A PROP3 users manual. RAE Technical Report 68294 (1968)
7	Y. Kozai	The motion of a close earth satellite. Astronom. J. 64, 367 (1959)
8	Y. Kozai	Effect of precession and nutation on the orbital elements of a close earth satellite. Astronom. J. 65, 621 (1960)
9	A. Cazenave F. Forestier F. Nouel J.L. Pieplu	Improvement of zonal harmonics by the use of observations of low inclination satellites, Dial, SAS and Peole. In *The use of artificial satellites for geodesy.* AGU Geophysical Monograph 15 (ed. S.W. Henrikson *et al.*) Washington DC (1972)
10	D.G. King-Hele G.E. Cook	Analysis of 27 satellite orbits to determine odd zonal harmonics in the geopotential. Planet. Space Sci. 22, 645 (1974) RAE Technical Report 73153 (1973)

REFERENCES (concluded)

No.	Author	Title, etc.
11	C.A. Wagner	Private communication to R.H. Merson (1971)
12	G.E. Cook M.D. Palmer	The synchronous mission analysis program SYNMAP. RAE Technical Report (to be issued)
13	D. Brouwer G.M. Clemence	Methods of celestial mechanics. Academic Press (1961)
14	S. Herrick	Astrodynamics Vol.II. Van Nostrand Reinhold Company, London (1972)
15	L. Lapidus J.H. Seinfeld	Numerical solution of ordinary differential equations. Academic Press, New York and London (1971)
16	L.G. Jacchia	Static diffusion models of the upper atmosphere with empirical temperature profiles. Smithsonian Contributions to Astrophysics 8, 215 (1965)
17	R.H. Gooding	A PREP users' manual. RAE Technical Report 69104 (1969)
18	A.W. Odell R.H. Merson	The SKYNET2 ephemeris program SKEPH2. RAE Technical Report 75084 (1975)
19	R.H. Merson	The dynamic model for PROP, a computer program for the refinement of the orbital parameters of an earth satellite. RAE Technical Report 66255 (1966)
20	-	Interpolation and allied tables. HMSO

Reports quoted above are not necessarily available to members of the public or commercial organisations.

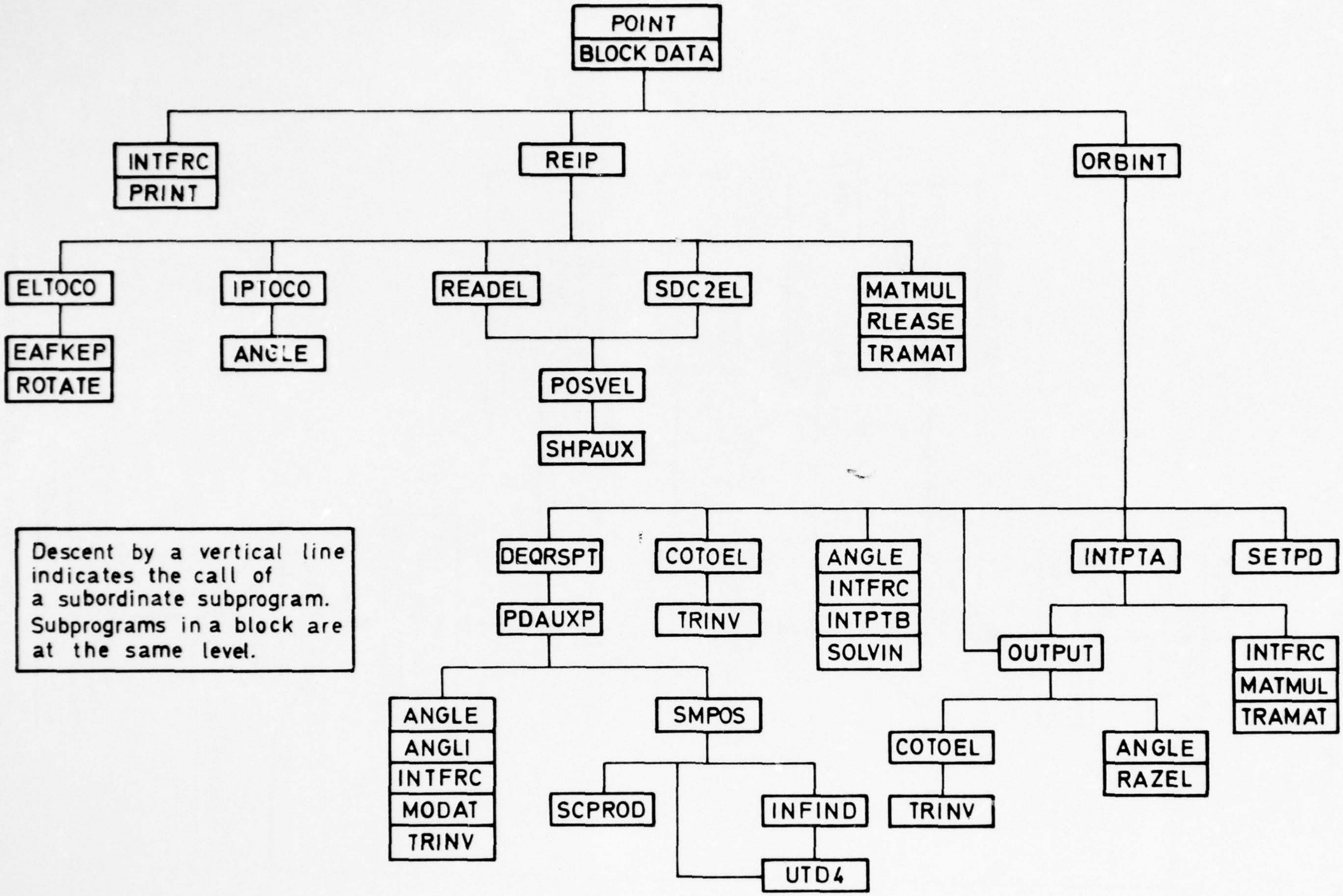

Fig.1 Calling structure for POINT program units

Fig.1

Fig.2

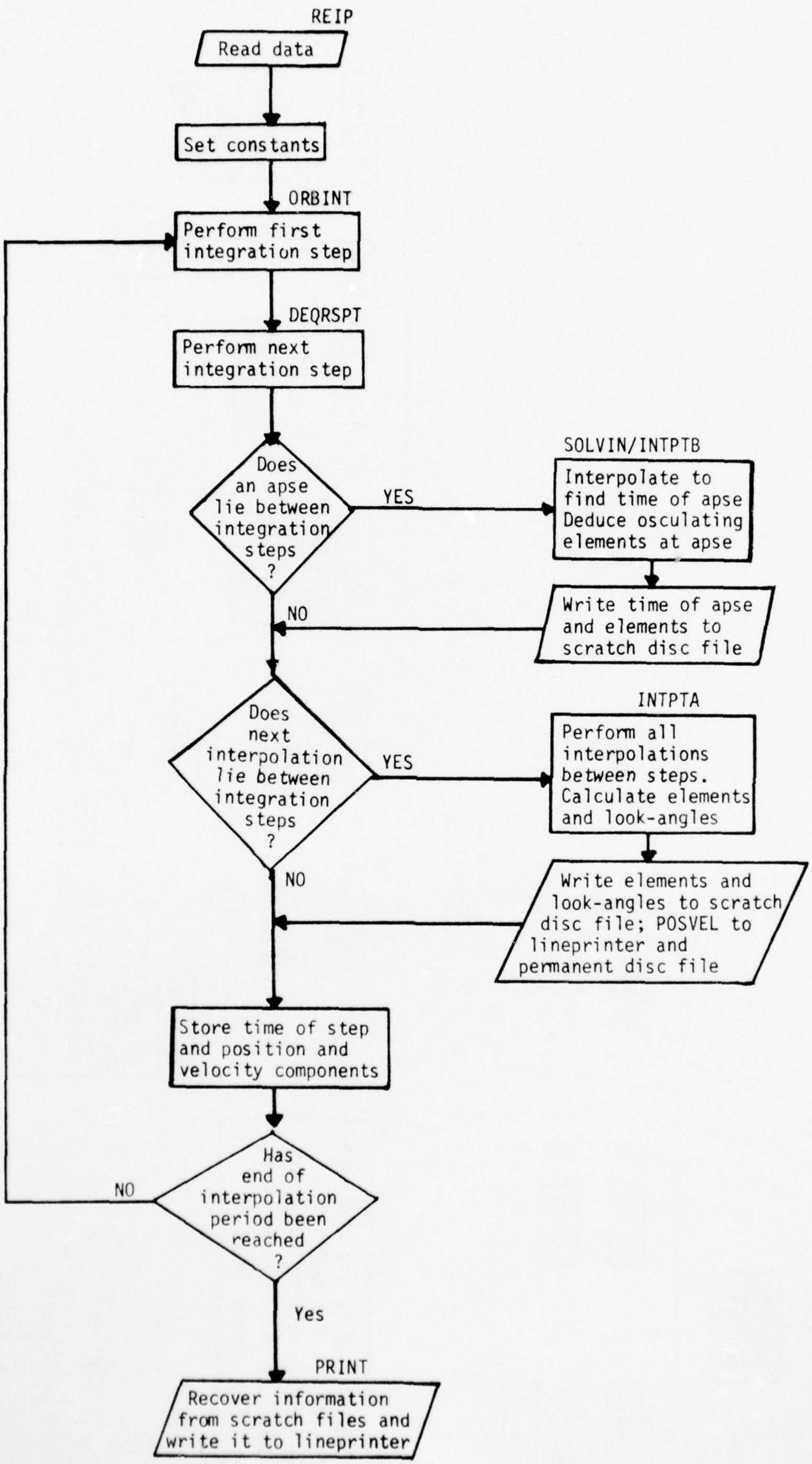

Fig.2 Simplified flow chart for POINT if all options exercised

TR 76129

ELLIPTIC ORBIT INTEGRATION

DRAG IS INCLUDED LUNI-SOLAR PERTURBATIONS NOT INCLUDED SOLAR RADIATION PRESSURE NOT INCLUDED

TIME OF FIRST INTERPOLATION FROM EPOCH = 2 HRS 0 MINS 0.000 SECS INTERPOLATION INTERVAL = 120.0 MINS

POSVEL

MJD	TIME	X (KM)	Y (KM)	Z (KM)	R (KM)	XDOT (KM/S)	YDOT (KM/S)	ZDOT (KM/S)	V (KM/S)
43445	0.0000000	5503.815	3554.292	391.579	6563.363	-4.7078370	7.8075802	-4.6936834	10.2553148
43445	0.0833333	-28584.400	-1757.670	-8727.432	29938.694	-2.2984387	-2.2300172	0.1383008	3.2054522
43445	0.1666667	-36672.369	-16205.709	-5585.676	40480.686	-0.1790358	-1.7071811	0.6275692	1.8276383
43445	0.2500000	-32746.156	-25651.148	-490.551	41599.672	1.2260489	-0.8643944	0.7516996	1.6764622
43445	0.3333333	-19188.085	-27379.601	4679.674	33759.806	2.5632877	0.5460647	0.6263875	2.6946227
43445	0.4166667	4122.868	-10769.923	5691.912	12860.287	3.3446496	5.7441658	-1.2072043	6.7556986
43445	0.5000000	-23477.636	2378.194	-8680.664	25143.767	-3.2268569	-2.2161213	-0.1679519	3.9181637

OSCULATING ELEMENTS

MJD	TIME	A (KM)	E	I	RA	AP	M	N (DEG/S)	R (KM)
43445	0.0000000	24667.522	0.73175203	27.5000	219.4461	172.5762	0.0000	0.0094516	6563.363
43445	0.0833333	24374.945	0.73062234	27.4873	219.3731	172.7079	68.4386	0.0095055	29938.694
43445	0.1666667	24374.596	0.73061048	27.4885	219.3716	172.7085	136.8840	0.0095057	40480.686
43445	0.2500000	24374.589	0.73060858	27.4888	219.3715	172.7078	205.3289	0.0095057	41599.672
43445	0.3333333	24374.864	0.73061404	27.4885	219.3713	172.7069	273.7744	0.0095056	33759.806
43445	0.4166667	24379.249	0.73070349	27.4828	219.3544	172.7230	342.2235	0.0095030	12860.287
43445	0.5000000	24373.669	0.73061156	27.4871	219.2105	172.9783	50.6567	0.0095063	25143.767

LOOK ANGLES FOR IOS

MJD	TIME	AZ	EL	RANGE (KM)	RANGE RATE (KM/S)
43445	0.0833333	106.8131	17.2378	27423.276	2.2241952
43445	0.1666667	96.4382	31.1450	36812.510	0.5578940
43445	0.2500000	84.6462	48.5031	36607.195	-0.5889486
43445	0.3333333	60.5140	58.7920	28141.853	-1.8184277
43445	0.4166667	53.7872	6.4001	10686.708	-1.6238959

LOOK ANGLES FOR GUAM

MJD	TIME	AZ	EL	RANGE (KM)	RANGE RATE (KM/S)
43445	0.0833333	223.2433	39.1012	25500.180	2.1545550
43445	0.1666667	242.5056	39.5619	36116.346	0.8610757
43445	0.2500000	259.4789	26.9038	38322.115	-0.2563460
43445	0.3333333	272.7695	13.4860	31698.663	-1.6965332
43445	0.4166667	294.7400	22.1383	9023.355	-5.6946500

LOOK ANGLES FOR NEW HAMPSHIRE

MJD	TIME	AZ	EL	RANGE (KM)	RANGE RATE (KM/S)
43445	0.5000000	151.1765	7.0679	23535.037	2.6430556

LOOK ANGLES FOR VANDENBERG

MJD	TIME	AZ	EL	RANGE (KM)	RANGE RATE (KM/S)

TABLE OF APSES

MJD	TIME	A (KM)	E	I	RA	AP	M	N (DEG/S)	R (KM)
43445	0.2191418	24374.571	0.73060846	27.4888	219.3715	172.7081	180.0003	0.0095058	42182.843
43445	0.4383218	24465.656	0.73173033	27.5005	219.2826	172.8470	359.9999	0.0094527	6563.394

LAST INTEGRATION STEP

43445 0.5002154956
-23537.6125082950 2336.9222894391 -8683.7530084835 -3.2159161064 -2.2172185900 -0.1639102417

Fig.3 Sample lineprinter output

Fig.4

`43445 0.0 12.0 120.0 0.0`	Time card
`3 0 0 1 1 1 4 0`	Control cards
`0.0 0.00461689 150.0 150.0`	
`-4.6667 55.4833 0.0 IOS`	Station cards
`13.6167 144.85 0.0 GUAM`	
`42.95 288.3667 0.0 NEW HAMPSHIRE`	
`34.8167 237.5 0.0 VANDENBERG`	
`24467.522 0.731752034 27.5000 219.4461 172.5762 0.0000`	Osculating elements

`43445 0.0 12.0 120.0 0.0`	Time card
`1 1 0 1 1 1 1 1`	Control cards
`0.0 0.00461689 150.0 150.0`	
`10.25531430 0.0000 117.3029 6563.363 3.4204 356.0000`	Injection parameters
` 1.68459822E+02-2.36945325E-01 4.34254639E-01`	Covariance matrix of injection parameters
`-1.05337572E+04 1.82701771E-02-2.91304240E-02`	
`-2.36945325E-01 1.13359136E-02-3.73616831E-06`	
` 2.34053026E+01-1.32153668E-03 2.29848297E-03`	
` 4.34254639E-01-3.73616831E-06 1.06082481E-02`	
`-6.76526159E-01-6.08169840E-06-6.20077460E-05`	
`-1.05337572E+04 2.34053026E+01-6.76526159E-01`	
` 1.27337905E+07-5.26943999E+00 8.27598956E+00`	
` 1.82701771E-02-1.32153668E-03-6.08169840E-06`	
`-5.26943999E+00 2.40459017E-03-3.83190211E-03`	
`-2.91304240E-02 2.29848297E-03-6.20077460E-05`	
` 8.27598956E+00-3.83190211E-03 6.80415065E-03`	

`0 0.0 12.0 120.0 0.0`	Time card
`4 0 0 1 1 0 0 0`	Control cards
`0.0 0.01 70.0 70.0`	
`1 07399U 74 60 D 75235.20997143 .00075973 00000-0 00000-0 0 00590`	SDC two line elements
`2 07399 047.1215 024.5326 7261081 106.8067 338.2930 2.36048543 8893`	

Fig.4 Specimen data decks for POINT

REPORT DOCUMENTATION PAGE

Overall security classification of this page

UNLIMITED

As far as possible this page should contain only unclassified information. If it is necessary to enter classified information, the box above must be marked to indicate the classification, e.g. Restricted, Confidential or Secret.

1. DRIC Reference (to be added by DRIC)	2. Originator's Reference	3. Agency Reference	4. Report Security Classification/Marking
	RAE TR 76129	N/A	UNLIMITED

5. DRIC Code for Originator	6. Originator (Corporate Author) Name and Location
850100	Royal Aircraft Establishment, Farnborough, Hants, UK
5a. Sponsoring Agency's Code	**6a. Sponsoring Agency (Contract Authority) Name and Location**
N/A	N/A

7. Title The elliptic orbit integration program POINT

7a. (For Translations) Title in Foreign Language

7b. (For Conference Papers) Title, Place and Date of Conference

8. Author 1. Surname, Initials	9a. Author 2	9b. Authors 3, 4	10. Date	Pages	Refs.
Cook, G.E.	Palmer, M.D.		Sept.1976	93	20

11. Contract Number	12. Period	13. Project	14. Other Reference Nos.
N/A	N/A		Space 512

15. Distribution statement
(a) Controlled by – [redacted]
(b) Special limitations (if any) –

16. Descriptors (Keywords) (Descriptors marked * are selected from TEST)
Transfer orbit. Elliptic orbits. Communications satellites. Synchronous satellites. Satellite lifetimes. Orbit integration.

17. Abstract

POINT is a computer program which evaluates the development of an earth satellite orbit by numerical integration. Provision is made for the inclusion of the following perturbations:

(i) earth's gravitational potential,
(ii) atmospheric drag, and
(iii) the gravitational attractions of the sun and moon.

A detailed description of the program is given, with full instructions for its use.

F5910/1

RAE Form A143